

COMPUMAGAZINE

LINUX

Manual de referencia

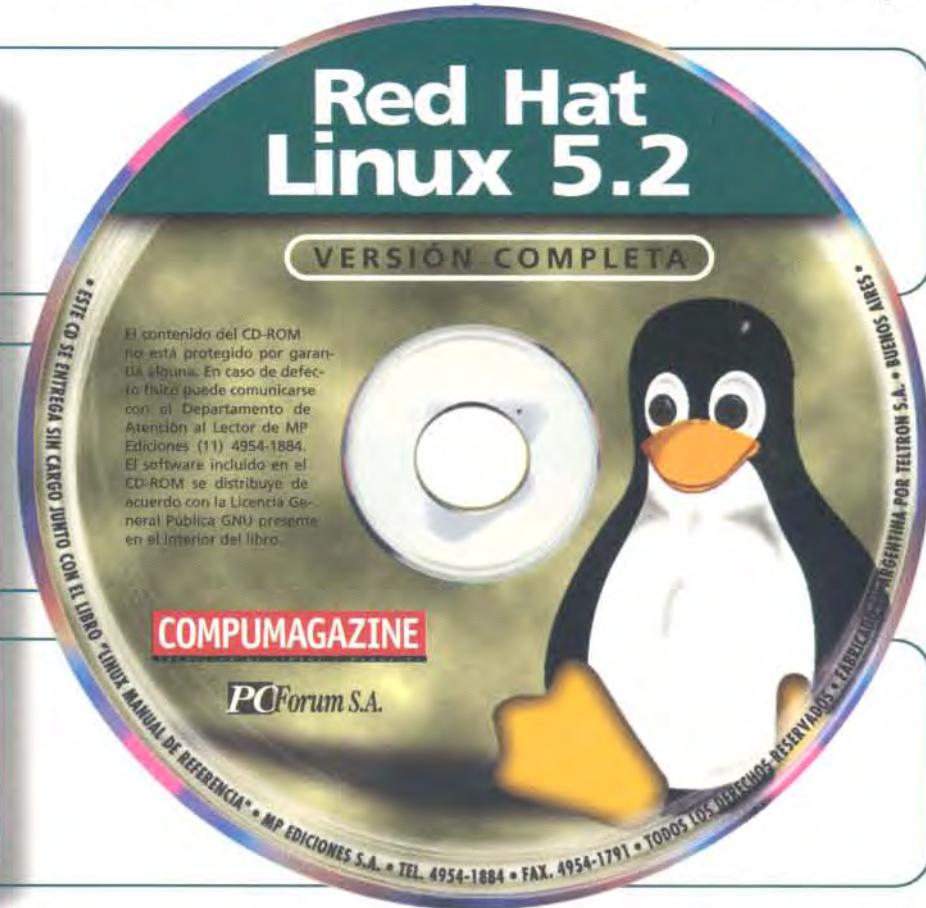
Domine a fondo el "otro" sistema operativo

Luis Tomás Wayar

- Instale en su PC la última versión de Red Hat Linux

- Maneje al 100% su configuración y operación

- Conozca y participe del efecto que temblar Microsoft



LINUX

Manual de Referencia

LUIS TOMAS WAYAR

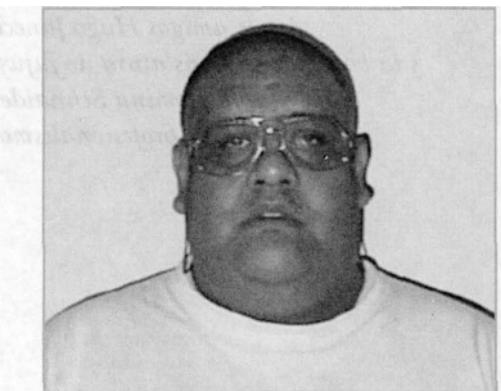
*A mis hijos, Luis y Melina,
mis más ambiciosos proyectos.*

*A mi esposa Elisa, por su incondicionalidad.
A mis hijos por el tiempo que me regalan.
A mi padre que me hizo conocer el hielo.
A mi madre que me hizo lo que soy.
A mi abuela Berta mi segunda madre.
A mis hermanos Paty, Seba y Javi
compañeros de la vida.
A mis tíos Mirta y Julián y mi primo Tincho
por una infancia feliz.
A mis amigos Hugo Janeck
y la barra de la Legislatura de Jujuy.
A mi editora Romina Schnaider
por su profesionalismo.*

Sobre el autor

Luis Tomás Wayar tuvo su primer contacto con una computadora a los 14 años, cuando su padre le regaló una TI994A; de ese acontecimiento ya pasaron dieciséis años. Con el correr del tiempo y la consecuente evolución de la informática, se dedicó a la programación de juegos. Es autor, entre otros, de *Oráculo*, *Tetris y Vibora*, este último ganador del 2º premio del concurso de la revista PC Users de Programación del Juego más Corto, en la categoría Lenguaje C.

Se desempeñó como Encargado del Centro de Cómputos de la Legislatura de Jujuy. Es Editor de la revista *El Bit Egoísta*, de distribución gratuita. En el año 1997, dictó un curso sobre LINUX y Sistemas Abiertos para los profesores de la Facultad de Ingeniería de la Universidad Nacional de Jujuy. Actualmente, se desempeña como empleado legislativo encargado de implementar la informatización de las tareas parlamentarias, y es Asesor de Seguridad Informática y LINUX de Imagine S.R.L., NEXT Servicios y Sistemas, y NOVA Informática. Es colaborador de las revistas PC Users de Argentina y JJF de España (de libre distribución).



PRÓLOGO

Desde que conocí el paradigma del Software Libre, estuve pensando en la manera de contribuir a la causa GNU. Comencé difundiendo LINUX entre mis allegados. Cuando creí que me encontraba en un buen estado general de conocimiento sobre el tema, me animé y empecé a colaborar con traducciones de documentación en inglés y, por último, traduje y modifiqué software de LINUX. En la actualidad, estoy abocado a un Infobot programado en PERL y deseo comenzar un proyecto para desarrollar un traductor de inglés para LINUX.

A pesar de esto, me quedaba una asignatura pendiente: escribir un libro sobre LINUX. Por fortuna, me llegó la oportunidad de la mano de Miguel Lederkremer, Director Editorial de MP Ediciones, quien me propuso encarar este proyecto.

He intentado hacer un libro útil, tanto para quien usa LINUX por primera vez como para quien ya lo conoce y desea reafirmar y ampliar conocimientos. Está orientado a lograr un aprendizaje progresivo a través de su lectura y sirve también como guía de referencia. En el libro se contemplan todos los temas necesarios para que el lector llegue a ser un usuario eficiente de LINUX. Sin embargo, en ocasiones no hemos entrado en detalles muy específicos para no distraer al lector, y este hecho no debe confundirse con un error, aunque, seguramente, los usuarios avanzados de LINUX encontrarán algunos a lo largo de estas páginas. Es la naturaleza falible del ser humano, por lo que todas las críticas y sugerencias serán bienvenidas.

Espero sinceramente que este libro sirva para que cada vez más gente conozca las virtudes de LINUX y del proyecto GNU.

Cómo contactar al autor

Puede enviar sus comentarios y sugerencias directamente al autor:

letal@iname.com

También puede escribir a MP Ediciones:

libros@mponline.com.ar

ÍNDICE

| | |
|--|-----------|
| Agradecimientos | 5 |
| Sobreelautor | 6 |
| Prólogo | 7 |
| Índice | 8 |
| CAPÍTULO 1. INTRODUCCIÓN | 13 |
| Linus Torvalds | 15 |
| Qué es LINUX | 15 |
| UNIX | 16 |
| Historia de LINUX | 17 |
| Anuncio y lanzamiento oficial | 17 |
| Repercusiones y adhesiones al proyecto | 18 |
| CAPÍTULO 2. CONCEPTOS PRELIMINARES | 21 |
| Sistema Operativo | 23 |
| Arquitectura | 24 |
| Dispositivos o devices | 26 |
| Tipos de dispositivos | 26 |
| Características | 26 |
| CAPÍTULO 3. INSTALACIÓN | 29 |
| Red Hat | 31 |
| Requerimientos | 31 |
| Conceptos previos | 32 |
| Información | 33 |
| Hardware | 34 |
| Red | 34 |
| Particiones | 35 |
| Booteo LINUX | 36 |
| Particiones | 39 |
| Formateo de particiones | 41 |
| Selección de paquetes | 42 |
| Configuración | 43 |
| Mouse | 43 |
| XWindow | 44 |
| Red | 45 |

| | |
|--|-----------|
| Huso horario | 45 |
| Servicios | 46 |
| Impresora | 46 |
| Clave de root | 47 |
| LILO | 47 |
| Reiniciando el sistema | 48 |
| CAPÍTULO 4. USUARIOS | 49 |
| Conceptos | 51 |
| El archivo PASSWD | 52 |
| Aregar un usuario | 54 |
| Borrar un usuario | 54 |
| Grupos | 55 |
| CAPÍTULO 5. COMPONENTES | 57 |
| Cuentas | 59 |
| Sesión | 59 |
| Terminal | 59 |
| Arranque de LINUX | 60 |
| init y su descendencia | 60 |
| Login y password | 61 |
| Entrada y salida | 61 |
| Terminales virtuales | 62 |
| Shell | 62 |
| Variables de entorno | 63 |
| Prompt | 64 |
| Midnight Commander | 66 |
| Aplicaciones | 66 |
| CAPÍTULO 6. COMANDOS | 69 |
| Formato | 71 |
| Comandos de archivos | 72 |
| Comandos generales | 78 |
| Comandos de información | 80 |
| Comandos de usuarios | 82 |
| Comandos de red | 82 |
| CAPÍTULO 7. SISTEMA DE ARCHIVOS | 85 |
| Universo LINUX | 87 |
| I-nodo | 87 |

| | |
|--|------------|
| Sistemas montados | 88 |
| Estructura | 89 |
| Permisos | 91 |
| Interpretación de permisos | 93 |
| Dependencias de permisos | 93 |
| Cambiando permisos | 94 |
| CAPÍTULO 8. PROCESOS | 97 |
| Definición | 99 |
| Tabla de procesos en UNIX | 99 |
| Creación de procesos | 101 |
| Control de tareas | 103 |
| Detención y relanzamiento | 106 |
| CAPÍTULO 9. APLICACIONES | 109 |
| Paquetes | 111 |
| Paquetes TGZ | 111 |
| Paquetes DEB | 111 |
| Paquetes RPM | 112 |
| Usando RPM | 113 |
| Compilación | 114 |
| CAPÍTULO 10. KERNEL | 117 |
| ¿Qué es el kernel? | 119 |
| ¿Por qué recompilar el kernel? | 119 |
| Los fuentes del kernel | 119 |
| Estables y experimentales | 120 |
| Preparación | 121 |
| Configuración | 121 |
| Code maturity level options | 122 |
| Loadable module support | 123 |
| General setup | 123 |
| Floppy, IDE and other blocks devices | 125 |
| Networking options | 127 |
| SCSI support | 127 |
| SCSI low-level drivers | 128 |
| Network device support | 128 |
| ISDN support | 130 |
| CD-ROM drivers | 130 |
| Filesystems | 130 |

| | |
|------------------------------------|------------|
| Character devices | 132 |
| Sound | 132 |
| Kernelhacking | 133 |
| Compilación | 133 |
| LILO | 134 |
| CAPÍTULO 11. REDES EN LINUX | 137 |
| Introducción | 139 |
| TCP/IP | 140 |
| Capas | 140 |
| Kernel | 142 |
| Configuración | 143 |
| Netcfg | 144 |
| Names | 144 |
| Hosts | 144 |
| Interfaces | 145 |
| Servidor de nombres | 147 |
| CAPÍTULO 12. XWINDOW | 149 |
| Definición | 151 |
| Conceptos | 152 |
| Servidores X | 152 |
| Manejadoresdeventanas | 153 |
| Comparaciones | 154 |
| FVWM-95 | 154 |
| AfterStep | 155 |
| Enlightenment | 156 |
| Window Maker | 157 |
| KDE | 158 |
| BlackBox | 159 |
| Emulador de terminal | 160 |
| Configuración de XWindow | 161 |
| CAPÍTULO 13. SEGURIDAD | 163 |
| Introducción | 165 |
| Planteo | 165 |
| Evaluación de riesgos | 166 |
| Evaluación de costos | 166 |
| Estrategia de protección | 167 |
| Análisis | 167 |

| | |
|--|-----|
| Evaluación de riesgos | 167 |
| Intrusos | 168 |
| Tipos de ataque | 169 |
| Ataques de software | 170 |
| Valor intrínseco y derivado de las áreas protegidas | 171 |
| Servidores de Internet | 172 |
| Definición de política y estrategia | 172 |
| Procedimiento para la detección de intrusos | 172 |
| Servidores | 174 |
| Medidas | 174 |
| APÉNDICE A. EJERCICIOS 177 | |
| Entrada en el sistema y consulta de información básica del sistema y de los usuarios | 179 |
| Comunicaciones con otros usuarios | 180 |
| Comandos básicos | 181 |
| APÉNDICE B. LICENCIA GENERAL PÚBLICA GNU 185 | |
| Preámbulo | 187 |
| Términos | 188 |
| APÉNDICE C. DICCIONARIO DE PALABRAS CASTELLANIZADAS 197 | |
| APÉNDICE D. LINUX EN INTERNET 203 | |
| Distribuciones | 205 |
| Software | 210 |
| Noticias | 213 |
| Varios | 217 |
| APÉNDICE E. ACERCA DEL CD-ROM 221 | |
| Bibliografía | 225 |
| Índice alfabético | 226 |

INTRODUCCIÓN

| | |
|--|----------|
| Linus Torvalds | .Pág. 15 |
| Qué es LINUX | .Pág. 15 |
| UNIX | .Pág. 16 |
| Historia de LINUX | .Pág. 17 |
| Anuncio y lanzamiento oficial | .Pág. 17 |
| Repercusiones y adhesiones al proyecto | .Pág. 18 |

“¿Quién hubiera pensado, tan sólo cinco años atrás, que un sistema operativo de gran calidad pudiera concretarse como por ensalmo a partir del trabajo aficionado y a tiempo parcial de varios miles de programadores esparcidos por todo el planeta y conectados tan sólo por las tenues hebras de Internet?”

La Catedral y el Bazar, Eric S. Raymond.

Linus Torvalds

Linus Benedictus Torvalds es un joven finlandés que, en el año 1991, tuvo la gran idea de hacer uno de los mejores Sistemas Operativos.

Actualmente, vive en los Estados Unidos y está casado con Patricia Miranda Torvalds, con quien tiene una pequeña hija. En la figura vemos a Linus Torvalds, un joven como cualquier otro.



Qué es LINUX

LINUX es un Sistema Operativo, tal como lo son Windows 98, DOS, Solaris, AIX e IRIX, por citar sólo algunos. Éste en particular se basa en UNIX, es decir que es una reimplementación del UNIX original de AT&T, aunque no tiene ni una sola línea de código en común con él.



WWW

La página oficial de LINUX es www.linux.org.

HAY QUE SABERLO

UNIX



HAY QUE SABERLO

EN RIGOR DE VERDAD

La primera versión de LINUX nunca pudo ser compilada con éxito.

En el año 1962, se emprendió un proyecto faraónico en lo que a informática se refiere. La idea era crear un centro de cómputos mundial, ubicado en Boston (EE.UU.), para lo cual habría que desarrollar un Sistema Operativo capaz de controlar semejante emprendimiento. Así nació MULTICS.

MULTICS llegó a funcionar, aunque fue demasiado ambicioso para la época. De él se rescataron muchas ideas que luego serían adoptadas por otros Sistemas Operativos.

Un empleado de Bell Labs, Ken Thompson, desarrolló un sistema en una DEC PDP-7, que, un poco en chiste, se llamó "UNICS", por su contraste con el enorme MULTICS. UNIX nace en el año 1970, de la mano de Ken Thompson y Dennis Ritchie, quienes trabajaban para los laboratorios Bell de AT&T. En un principio, UNIX estaba escrito en Assembler, pero no pasó mucho tiempo para que Dennis Ritchie, el inventor del lenguaje de programación C, reescribiera completamente el código de UNIX en este nuevo lenguaje.

Éste fue el primer gran salto de UNIX, porque gracias a estar escrito en C y a que AT&T otorgara licencias a varios colegios y universidades, se logró la portabilidad del Sistema Operativo, uno de los principales paradigmas de UNIX.

Una de las universidades más influyentes en el desarrollo de UNIX fue la Universidad de California, en Berkeley (UCB). En sus laboratorios y centro de cómputos se implementaron muchas de las mejoras de UNIX. Lamentablemente, esto también originó una fractura en las filas de los seguidores del sistema; AT&T, por su parte, continuó con el desarrollo, y los sistemas divergieron y se formaron dos corrientes para UNIX. Hoy en día, se está realizando un gran esfuerzo para subsanar estas dificultades, por lo que se habla de sistemas "convergidos", en los que la convergencia se refiere a tratar de corregir las diferencias entre el planteo de UNIX que hace la empresa AT&T y el enfoque de la Universidad de Berkeley.



DEFINICIONES

UNIX

UNIX está definido como un sistema operativo multiusuario de tiempo compartido.



MÁS DATOS

OTRA VERSIÓN

Existe otra versión libre de UNIX, llamada Free BSD.

Historia de LINUX

Originalmente, comenzó como un proyecto de Linus, quien siendo estudiante de la Universidad de Helsinki, en Finlandia, trabajó sobre una versión de Minix escrita por Andy Tenembaum para lograr un sistema capaz de correr en computadoras con microprocesadores de la familia Intel X86. En un principio, Linus escribió WNUX en Assembler, pero luego, llegada la hora de escribir algunos drivers, comenzó a utilizar C, con lo que notó una importante aceleración en los tiempos de desarrollo.



DEFINICIONES

DRIVER

Software encargado de controlar y administrar algún dispositivo.

A fines de agosto de 1991, Linus ya tenía una precaria versión de LINUX, la 0.01, que era capaz de montar en disquetes y contaba con un pequeño sistema de archivos. Esta versión no fue anunciada en forma oficial, ya que se necesitaba contar con Minix para poder compilarla.

Anuncio y lanzamiento oficial



HAY QUE SABERLO

DIFUSIÓN DEL PROYECTO

Internet jugó un papel preponderante en la difusión y posterior vinculación de los interesados del proyecto LINUX.

Dos meses después, el 5 de octubre de 1991, LINUX, el incipiente Sistema Operativo, ya era capaz de ejecutar un shell (bash) y el compilador GCC. En la lista de noticias **comp.os.minix**, Linus anuncia su criatura con las siguientes palabras:

"Si suspiras al recordar aquellos días, cuando los hombres eran hombres y escribían sus propios drivers. ¿Te sientes sin ningún proyecto interesante y te gustaría tener un verdadero Sistema Operativo que pudieras modificar a placer? ¿Te resulta frustrante contar únicamente con Minix? Entonces este artículo es para vos. Como dije hace un mes, estoy trabajando en una versión gratuita de algo así como Minix para computadoras AT 386. He alcanzado la etapa en la que puede ser utilizable y voy a poner los fuentes para su distribución."

Repercusiones y adhesiones al proyecto



HAY QUE SABERLO

LINUX, EL PREFERIDO DE LOS HACKERS

Es común escuchar decir que LINUX es el sistema operativo favorito de los hackers. Esto se debe, principalmente, a que los programas se proveen con sus fuentes y a su calidad de sistema operativo de red.

Inmediatamente después de liberar la primera versión oficial, comenzaron las repercusiones, en especial en el mundillo de los gurúes de UNIX y en los hackers. No tardaron en llegar innumerables adhesiones en forma de consejos, aportes y ofrecimientos para sumarse al equipo de programadores.

Es así como, a partir de un proyecto aparentemente insignificante de un estudiante finlandés, se originó toda una movida dentro del mundo de la informática que hizo que, con el correr de los años, hasta las mayores empresas de desarrollo de sistemas operativos comenzaran a temblar.



DEFINICIONES

GNU

Es una frase recursiva que significa GNU is not UNIX.

Afortunadamente, LINUX había adherido desde un principio su proyecto a la licencia GNU o GPL, lo que significa que todo el sistema

operativo es de libre distribución, incluidos sus fuentes. Esto permite que otros programadores puedan modificar el código para mejorarlo o corregirlo.

Con semejante repercusión, LINUX estaba destinado a convertirse en un grande. Al principio, todo fue muy difícil, dado que, por las características del proyecto, no se contaba con ningún tipo de apoyo empresarial. Por suerte, cuando la Universidad de Helsinki observó el impacto producido por LINUX, desafectó a Linus de sus estudios y le otorgó una dedicación exclusiva al emprendimiento.



DEFINICIONES**FUENTE**

Es el código original de un programa sin compilar. Es el texto escrito por el programador.

Poco a poco, comenzaron a sumarse experimentados programadores y LINUX empezó a tomar forma como un completo sistema operativo. Una a una, se fueron agregando todas las utilidades y no se tardó en lograr los primeros editores clónicos de los propios de UNIX, compiladores, juegos, aplicaciones y hasta un sistema de ventanas.

CONCEPTOS PRELIMINARES

| | |
|------------------------------|----------|
| Sistema Operativo | .Pág. 23 |
| Arquitectura | .Pág. 24 |
| Dispositivos o devices | .Pág. 26 |
| Tipos de dispositivos | .Pág. 26 |
| Características | .Pág. 26 |

"El público estaba enfervorizado. Torvalds representaba para ellos algo así como la presencia de Jean-Paul Sartre en Saint-Germain-des-Prés. Si Sartre había encendido las pasiones de los jóvenes de París, Torvalds emocionaba hasta el paroxismo a sus seguidores de Silicon Valley."

Diario El Mundo, sobre la llegada de Linus a San Francisco, California

Sistema Operativo



MÁS DATOS

FUENTES DEL KERNEL

Los fuentes del kernel de LINUX están en www.kernel.org.

Un sistema operativo está compuesto principalmente por un núcleo o kernel; un recubrimiento para éste, llamado shell o intérprete de comandos, que hace las veces de intermediario entre el usuario y el núcleo; y una colección de utilidades para el correcto uso y administración del sistema.

Podemos definir un sistema operativo como una máquina virtual amigable para el ser humano que recubre una máquina real (el hardware de la computadora) que es hostil para el hombre. Así, máquinas diferentes pueden resultar similares para el usuario si utilizan el mismo sistema operativo.



HAY QUE SABER

INTERESANTE

En DOS, el shell es el famoso **command.com**; en LINUX, el más popular es **bash**.

Sin pretender ser muy extensos y específicos, vamos a intentar definir qué es un sistema operativo citando sus principales funciones.

Para comenzar, podemos decir que sus tareas fundamentales son las siguientes:

- Ejecución de programas.
- Administración de recursos del sistema.
- Interfase primaria entre el usuario y el sistema.

- Soporte para aplicaciones.
- Protección del sistema y sus recursos.
- Gestión de usuarios.
- Provee las herramientas básicas para el mantenimiento del sistema.



[MÁS DATOS](#)

DOS

DOS nace incorporando algunos elementos de CPM y de UNIX. Si bien fue un sistema operativo muy popular en los años '80 y principios de los '90, en la actualidad muy poca gente lo usa.

Éstas son, básicamente, las funciones de un sistema operativo; todos deben cumplir, por lo menos, con estas premisas. Pero como sabemos, existen muchas formas de hacer lo mismo, y por eso nos encontramos con un gran abanico de posibilidades: DOS, CPM, Windows, OS/2, VAX, UNIX, etc. Todos éstos son sistemas operativos y todos hacen en esencia prácticamente lo mismo. Pero como veremos ahora, no todos lo hacen igual y no todos tienen las mismas prestaciones.

Arquitectura

Al ser un clon de UNIX, LINUX usa como punto de partida su misma arquitectura. Ésta se basa en un planteamiento nuclear, es decir que la estructura principal se asemeja a la estructura de un átomo. Más precisamente, podemos visualizar la estructura como las capas de una cebolla, en la que contamos con un kernel o núcleo, alrededor del cual hay un shell o capa de interfase con el usuario. Si bien esto no es tan simple, comenzaremos por definirlo de esta manera para luego entrar en mayores detalles. Veamos en la **Figura 1** cómo es esta estructura.

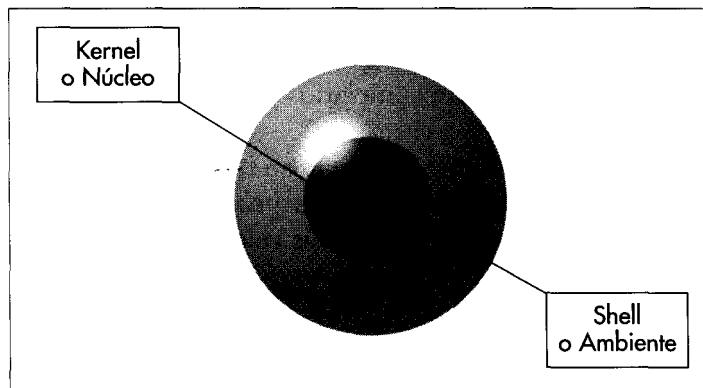


Figura 1. Estructura básica de LINUX.

Si bien ésta es la estructura global de la arquitectura UNIX-LINUX, en realidad, este átomo cuenta con varios tipos de electrones en su capa externa, siguiendo con la analogía (**Figura 2**). Por ahora, sólo definiremos cuáles son y, más adelante, los veremos en profundidad.

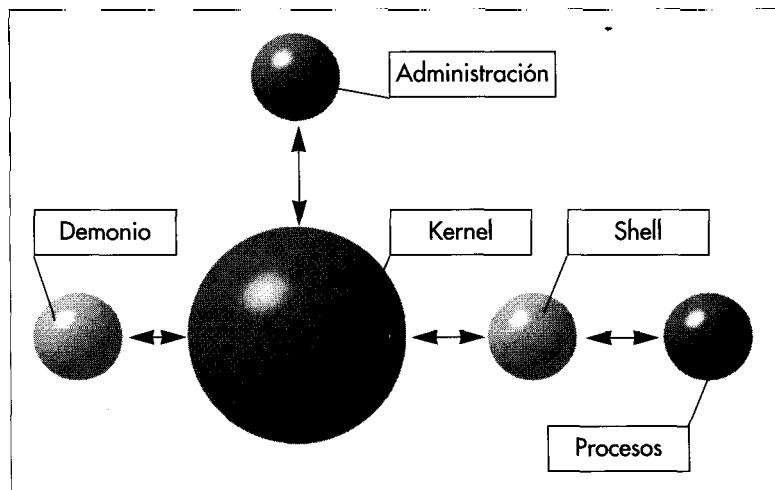


Figura 2. Estructura de electrones de la capa externa del átomo.

- El núcleo (Kernel) es el encargado de las funciones esenciales:
 - Control de interrupciones.
 - Control de procesos.
 - Control de dispositivos (devices).
 - Control de los sistemas de archivos.
- Capa "burocrática" de procesos administrativos.

- Capa de los "demonios".
- Capa de los "shells" que interactúan con el usuario.
- Capa de procesos iniciados por los usuarios a través del shell.

Finalmente, debemos decir que, dado que LINUX fue creado por programadores y pensado para programadores, en él encontraremos ventajas y desventajas. Entre las ventajas, contamos con innumerables herramientas a la hora de programar; como desventaja, la sintaxis es un poco críptica.

Dispositivos o devices

En LINUX, todos los dispositivos conectados a la CPU son considerados y tratados como archivos. A su vez, estos dispositivos-archivos son controlados por *device-drivers*.

Tipos de dispositivos

Los dispositivos se clasifican en:

- 1. Seriales:** en un dispositivo serial, se lee o escribe carácter por carácter. Ejemplos: teclado, mouse, módem.
- 2. Bloque:** este tipo de dispositivos está dividido en bloques, y se puede leer o escribir cualquier bloque al azar, es decir que el acceso es aleatorio. Ejemplo: disco rígido, disquetes, memoria.
- 3. Especiales:** estos dispositivos cubren casos como pantallas, dispositivos gráficos, etc.

Características



HAY QUE SABERLO

ACCESO GARANTIZADO

LINUX usa **multitarea preferente**, la que garantiza acceso a los recursos para todos los procesos en ejecución.

- Multitarea real: la multitarea es una virtud codiciada por muchos sistemas operativos, pero lograda por muy pocos. Casi todos los que dicen serlo lo son a medias; en cambio, la multitarea que usa LINUX administra los recursos garantizando que todos los procesos en ejecución, incluidos los que corren en background, tengan su tiempo de procesador.
- Multiusuario: LINUX es un sistema operativo multitarea y, obviamente, debe ser multiusuario, es decir que permite a distintas personas acceder al sistema compartiendo los recursos que el Administrador del Sistema (superusuario,root) le asigne de acuerdo con su jerarquía, dándole además a cada uno la privacidad y protección necesarias.
 - Soporta los estándares POSIX, BSD, IEEE y System V.
 - Es capaz de operar con todos los sistemas de ficheros estándar, como FAT de DOS, VFAT de Windows, OS2/FS, ISO9660 (CD-ROM), Minix y otros.
- Es un sistema operativo de red. Su protocolo es TCP/IP y acepta todas las placas de red, SLIP, PPP para acceso serial a TCP/IP, PLIP para conexiones en paralelo, NFS Sistema de Archivos de Red, Telnet, NNTP, SMTP, IPX, AppleTalk, Samba para compartir archivos e impresoras con Windows 95.
- Es de 32 bits reales en una PC y de 64 en una Alpha. Su kernel utiliza las características del modo protegido de los micros 386 y superiores.
- Tiene soporte completo de hardware, multimedia, módems, impresoras, placas de video, monitores, teclados, mouse, etc.
- Cuenta con un poderoso entorno gráfico con innumerables sistemas de ventanas. Los más populares son FWWM, GNOME, KDE, CDE, Enlightenment, Afterstep, NextLevel, TWM, etc.
- G.N.U.: si bien Linus Torvalds es el titular del derecho de autor de LINUX, todo, incluido el kernel con sus fuentes, está bajo licencia GNU. Esta licencia permite que todo el software desarrollado bajo este concepto sea de libre distribución, de modo que se ofrece software de calidad al público en general sin que los programadores pierdan sus derechos de autor y dejando abierta la posibilidad para que cualquier otro programador pueda ampliar o modificar el programa.
- Librerías compartidas: gracias a esta característica, no es necesario que las rutinas contenidas en librerías estándar se carguen más de una vez en memoria, ya que cualquier programa que se encuentre en tiempo de ejecución puede acceder a ellas. De esta manera, los binarios (ejecutables) de LINUX son de menor tamaño y permiten ahorrar espacio en disco y memoria.

- Carga por demanda: en la mayoría de los sistemas operativos, cada nuevo proceso (aplicación) que se lanza demanda memoria y recursos, y cuando se termina la RAM, automáticamente se comienza a hacer uso de la memoria virtual. Esto no ocurre en LINUX, puesto que su administración inteligente de memoria pasa a memoria virtual cualquier proceso que se encuentre activo pero no en uso, con lo que se garantiza RAM para las aplicaciones activas y en uso.
- Administración de memoria: todos los procesos tienen garantizada una zona protegida de memoria para su ejecución, sin que el mal funcionamiento de una aplicación cuelgue todo el equipo.
- Redes TCP/IP: TCP/IP es un protocolo propio de los sistemas UNIX luego heredado por LINUX. Contamos directamente con él en el kernel.
- Aplicaciones: gracias a la licencia GNU, el caudal de aplicaciones disponibles para LINUX crece a un ritmo vertiginoso, especialmente en Internet. Podemos decir que existe software para casi todas las necesidades. La única limitación para los usuarios de habla hispana es el idioma, ya que es muy difícil encontrar software traducido o de origen latino para LINUX.
- Desarrollo: LINUX es un sistema operativo hecho y pensado por programadores para programadores. A partir de esto, podemos encontrar compiladores o intérpretes para todos los lenguajes existentes. Algunos ejemplos son los siguientes: C (propio de los sistemas UNIX), C++, ADA, PASCAL, PERL, MODULA 2, FORTRAN, PHYTON, EIFFEL, BASIC, RPG, SMALLTALK, COBOL, ASSEMBLER, OBERON, etc.

INSTALACIÓN

| | |
|-------------------------------|----------------|
| Red Hat | Pág. 31 |
| Requerimientos | .Pág. 31 |
| Conceptos previos | .Pág. 32 |
| Información | .Pág. 33 |
| Hardware | .Pág. 33 |
| Red | .Pág. 34 |
| Particiones | .Pág. 35 |
| Booteo LINUX | .Pág. 36 |
| Particiones | .Pág. 39 |
| Formateo de particiones | .Pág. 41 |
| Selección de paquetes | .Pág. 42 |
| Configuración | .Pág. 43 |
| Mouse | .Pág. 43 |
| XWindow | .Pág. 43 |
| Red | .Pág. 45 |
| Huso horario | .Pág. 45 |
| Servicios | .Pág. 46 |
| Impresora | .Pág. 46 |
| Clavederoot | .Pág. 47 |
| LILO | .Pág. 47 |
| Reiniciando el sistema | .Pág. 48 |

“Dejen en paz a Bill Gates, porque la auténtica amenaza es Linus Torvalds... Vayan a decirles a esos socialistas de la programación que se lleven a Europa sus concepciones radicales del desarrollo cooperativo del “free code”. Los americanos exigen su libertad de pagar a sus creadores lo que es justo por los programas que dominan el mercado.”

Diario San Francisco Chronicle

Red Hat



HAY QUE SABERLO

WWW

La página oficial de Red Hat se encuentra en www.redhat.com.

Red Hat, en inglés, significa Sombrero Rojo y es el nombre de una de las muchas distribuciones de LINUX disponibles en la actualidad. Ésta en particular nace en el verano estadounidense de 1994 en Carolina del Norte, cuando un reducido grupo de entusiastas del sistema deciden crear una instalación de LINUX poniendo énfasis en la posibilidad de seleccionar los componentes que se van a instalar, de poder actualizarlos sin necesidad de tener que reinstalar todo el sistema e incluso de poder incorporar nuevos programas. Esto es lo que se conoce como paquetes y se trata de archivos con extensión .rpm (Manejador de Paquetes Red Hat).

Requerimientos



MÁS DATOS

PRECAUCIONES PARA LA INSTALACIÓN

No intenten instalar LINUX a medida que lean por primera vez este capítulo; es conveniente leerlo primero íntegramente y luego poner manos a la obra.

En primer lugar, debemos conseguir los CD-ROMs de instalación, pidiéndolos directamente a la casa central en los Estados Unidos o comprándolos en cualquier negocio de venta de software. En cualquiera de estos casos, tendremos dos CDs, uno con la distribución binaria de los paquetes (éste es el que usaremos para instalar) y otro con los fuentes de los programas que componen la instalación.

Conceptos previos

Antes de comenzar, debemos conocer algunos conceptos que serán muy útiles durante la instalación. En LINUX, a diferencia de DOS/WIN, la barra de directorios es / y no \.

| DOS / Win | LINUX |
|-----------|-------|
| com1 | ttys0 |
| com2 | ttys1 |
| com3 | ttys2 |
| com4 | ttys3 |

Equivalentias entre los nombres de las salidas serie en DOS/Windows y LINUX.

Aclaración: a partir de la versión 2.2.0 del kernel los cuax han quedado obsoletos.

LINUX utiliza un sistema de control de periféricos que asigna a cada uno de ellos un fichero virtual, que depende del subdirectorio /dev/. Por ejemplo, todos los discos rígidos de norma IDE se denominan con **hd** seguido de una letra que denota el controlador al que está conectado y si se trata de un disco maestro o esclavo; finalmente, un número nos indica la partición con la que estamos tratando. De esta manera, **hdal** es la primera partición del primer disco rígido conectado a la primera controladora IDE como maestro. Para entenderlo mejor, veamos un ejemplo.

Imaginemos la siguiente configuración de una PC en DOS/WIN: dos discos rígidos IDE y una lectora de CD; el primer disco rígido está conectado a la primera controladora IDE como maestro y tiene dos particiones: C y D; el segundo disco está como esclavo de la primera controladora y sólo tiene una partición, conocida en DOS/WIN como E; por último, tenemos una lectora de CDs conectada a la segunda controladora como maestro, que en DOS/WIN se llamaría F. En la **Figura 1** se presentan las equivalencias entre la nomenclatura de DOS y de LINUX.



HAY QUE SABERLO

IMPORTANTE

En caso de discos SCSI, la nomenclatura es /dev/sda1. Observe que se cambia hd de los discos IDE por sd.

| Dispositivo | Descripción |
|-------------|--|
| /dev/hda | Disco rígido conectado a la 1º controladora IDE como maestro |
| /dev/hda1 | Primera partición. Normalmente, C:\ |
| /dev/hda2 | Segunda partición en DOS/WIN: D |
| /dev/hdb | Disco rígido conectado a la 1º controladora IDE como esclavo |
| /dev/hdb1 | Primera partición del disco esclavo: E |
| /dev/hdc | Lectora de CD-ROM: 2º controladora maestro |

Figura 1. Equivalencias entre la nomenclatura DOS y LINUX.

Información

Durante la instalación, necesitaremos disponer de la información sobre las características de nuestro hardware y sobre la configuración del mismo.

Hardware

Ésta es una lista de la información que recopilaremos antes de comenzar la instalación.

- Discos rígidos: cantidad, norma (IDE o SCSI), tamaño, dónde está conectado y cuántas particiones tiene.
- CD-ROM Drive: lo más importante es la norma; en caso de no ser IDE ni SCSI, necesitaremos la marca y el modelo.
- Placa SCSI: marca y modelo.

- Memoria: sólo debemos saber la cantidad de RAM de la que disponemos.
- Placa de Red: marca, tipo, modelo, puerto de entrada y salida, y número de interrupción.
- Mouse: tipo, puede ser serie, PS/2, bus mouse, etc., protocolo (microsoft, logitech, mouseman, etc.), cantidad de botones (2 o 3) y el puerto al que está conectado.
- Teclado: tipo y distribución (español, inglés, etc.).
- Placa de video: marca, modelo y memoria.
- Monitor: marca, modelo y frecuencias de refresco horizontal y vertical (esto se encuentra en el manual del monitor; en caso contrario, se pueden usar frecuencias estándar).

Red

Información necesaria para la correcta configuración de la red bajo LINUX:



HAY QUE SABERLO

CONSULTAS IMPORTANTES

Si no conocemos estos datos, lo mejor es preguntarle al administrador de la Red, quien seguramente podrá informarnos al respecto.

- Dirección IP: es el número que usará el protocolo TCP/IP para identificar nuestra máquina. Normalmente, se trata de cuatro números no mayores que 255 separados por ". ". Por ejemplo, 127.0.0.1
- Máscara de la Red: también son cuatro números separados por puntos. Ejemplo: 255.255.255.0.
- Dirección IP de Gateway: otros cuatro números separados por puntos.
- Dirección IP del Servidor de Nombres de Dominio (DNS): pueden ser uno o más, iguales a los anteriores.
- Nombre de Dominio de la red: es el nombre con que se conoce a toda la red. Por ejemplo: inka.com.ar.
- Hostname: el nombre de nuestra PC en la Red. Por ejemplo, la mía se llama amankay.

Particiones

[MÁS DATOS](#)**FIPS**

FIPS no funciona con discos de FAT32.

Para crear una nueva partición donde instalar LINUX, no es necesario borrar nada ni perder datos, gracias a un programa llamado FIPS que nos permite particionar el disco sin ninguna pérdida. En caso de tener instalado algún sistema operativo previo, deberemos defragmentar el disco para liberar espacio al final del mismo.

Por lo menos, necesitaremos 500 MB para poder instalar LINUX y contar con espacio suficiente para maniobrar.

Vamos paso por paso:

- 1) Liberamos todo el espacio posible de nuestro disco rígido borrando lo que no necesitemos.
- 2) Agrupamos todo lo que nos quede instalado usando el Defragmentador de Disco de Windows 95 o defrag.exe si lo hacemos desde DOS/Windows 3.11.
- 3) Hacemos un backup de los datos importantes. Esto no es obligatorio; es muy bajo el riesgo de pérdida de datos.
- 4) Colocamos el disco 1 de Red Hat y entramos al subdirectorio \dosutils\.
- 5) Ejecutemos el programa FIPS.EXE, con el que podremos quitar espacio a la partición DOS/WIN para posteriormente instalar LINUX. Nos encontraremos con la presentación del programa y luego pasaremos a una ventana donde veremos un detalle del estado de nuestro rígido. Acto seguido, nos presenta información sobre nuestro sector de booteo, para finalmente permitirnos quitar espacio a nuestra partición DOS/WIN y crear una nueva partición DOS que luego borraremos para crear dos particiones LINUX, una nativa y otra de SWAP.
- 6) Por último, grabamos las modificaciones y booteamos para ver si todo salió bien.

Booteo LINUX

Si booteemos desde un disquete con DOS, podemos pasar directamente a leer el siguiente párrafo. Si, en cambio, estamos arrancando desde el rígido con Windows instalado, tendremos que pulsar F8 en el momento del arranque para seleccionar la opción Sólo símbolo del sistema, la cual deberá tener soporte para CD.

Ahora, con el CD 1 de Red Hat colocado en nuestra lectora de CDs, haremos lo siguiente:

Supongamos que a nuestro CD-ROM Drive se le asigna la letra F. Desde la línea de comando, ingresemos lo siguiente:

```
C:\>f:  
F:\>cd \dosutils  
F:\dosutils\autoboot.bat
```



HAY QUE SABERLO

ATENCIÓN

Esto se debe hacer desde DOS. Windows no se debe estar ejecutando.



MÁS DATOS

PARA TENER EN CUENTA

En PCs con BIOS nuevos, se puede arrancar directamente desde el CD.

A continuación, veremos un clásico booteo LINUX. Saldrán varias líneas de texto bastante ininteligibles para quien instala LINUX por primera vez, pero no se asusten, porque dentro de poco sabremos qué quiere decir cada una.

Ahora estamos frente a un cuadro de diálogo que nos pregunta si tenemos monitor color. En caso afirmativo, pulsamos Enter; si no, con Tab cambiamos a NO y pulsamos Enter (**Figura 2**).

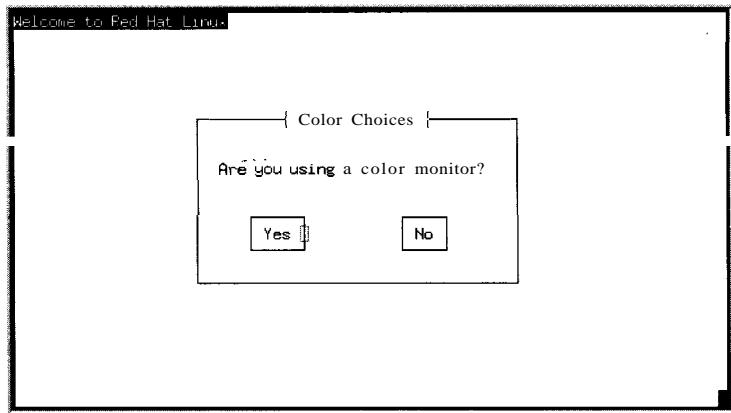


Figura 2. Seleccionemos el tipo adecuado de monitor

Luego de la presentación y bienvenida que pasaremos pulsando Enter, el programa de instalación nos preguntará acerca de la distribución de las teclas de nuestro teclado y nos presentará una lista con todas las posibilidades disponibles. Seleccionamos la adecuada usando las flechas de Arriba y Abajo. Podemos ver que existen más opciones que las que se ven en la **Figura 3**.

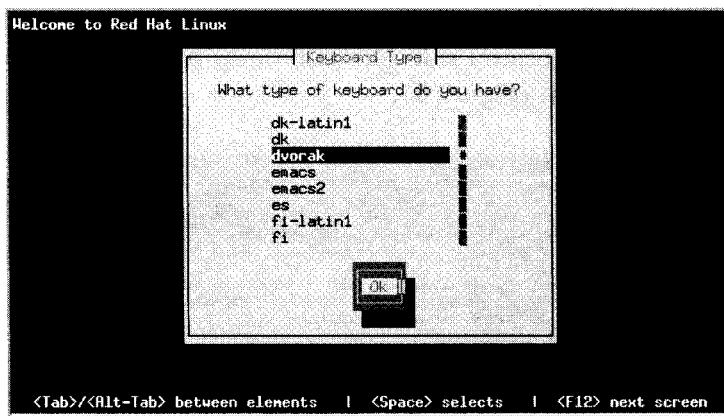


Figura 3. Son soportados tanto los teclados españoles como los latinoamericanos.

Nos preguntará si necesitamos soporte PCMCIA; contestemos según corresponda.

Luego nos pregunta sobre el método a usar para instalar LINUX,

que pueden ser CD-ROM, NFS, HARD DRIVE, FTP, como se ve en la **Figura 4**. En nuestro caso, seleccionaremos CD-ROM.

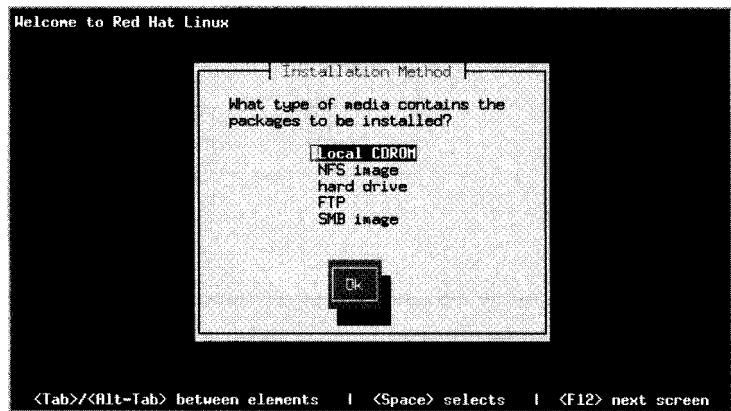


Figura 4. Mediante FPT o NFS, podremos instalarlo desde cualquiersitio de Internet.

Luego el sistema intentará reconocer con qué tipo de lectora contamos. En caso de fracasar, expondrá una serie de alternativas donde seguramente encontraremos la nuestra si es soportada por LINUX. Los casos más comunes son: IDE (ATAPI),SCSI (elige el controlador apropiado), OTHER si nuestro CD-ROM Drive es de una marca con norma propia, como algunos SANYO o Mitsumi.

Nos preguntará si lo que queremos hacer es una instalación o una actualización (**Figura 5**).

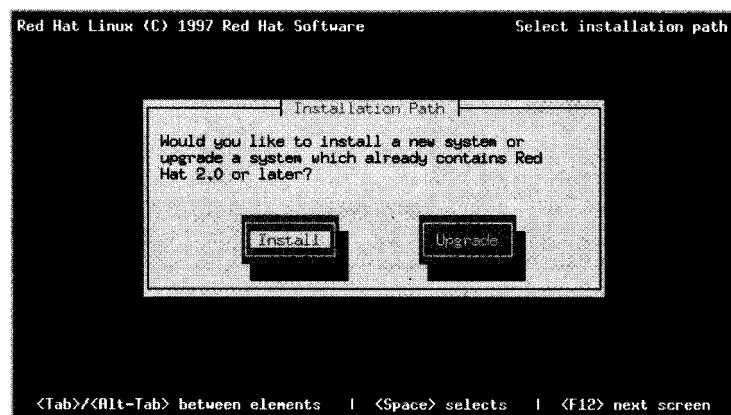


Figura 5. Si optamos por una actualización, se mantienen las configuraciones previas.

Luego el sistema intentará detectar si tenemos dispositivos SCSI; en caso de no detectarlos, puede ser que nos haga una pregunta, a la que responderemos según corresponda y elegiremos el controlador apropiado.

Particiones

Ahora estamos frente a un cuadro de diálogo donde debemos elegir entre dos utilidades para particionar el disco.



HAY QUE SABERLO

UN BUEN TAMAÑO PARA LA PARTICIÓN

Un buen tamaño para la partición de **SWAP** es el doble de la **RAM** instalada.

La primera es Disk Druid, una nueva herramienta muy fácil de usar e intuitiva (**Figura 6**).

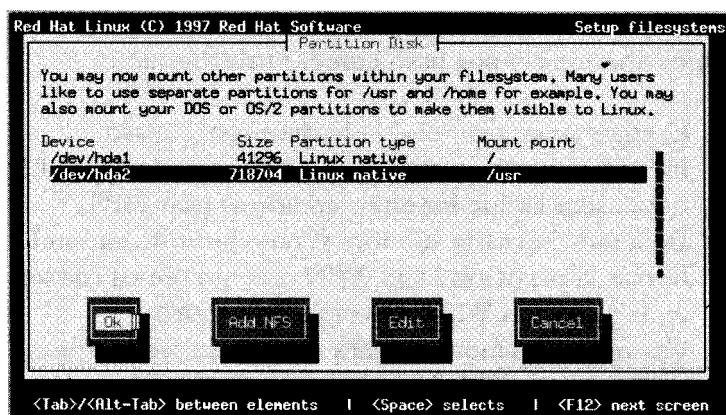


Figura 6. Disk Druid es una aplicación que se introdujo a partir de la versión 5.0 de Red Hat.

La segunda es el viejo y conocido fdisk pero en versión LINUX. Vamos a usar Fdisk, ya que de paso aprenderemos su manejo para futuras ocasiones. De todas formas, Disk Druid es mucho más fácil y no debería presentar mayores complicaciones. Fdisk es menos amigable y forma parte de las utilidades básicas de LINUX. Es un programa con una interfase con el usuario muy limitada; sin embargo, la flexibilidad y el poder que presenta son bastante buenos. Antes que nada, veamos un repaso de sus comandos, los cuales se ingresan directamente interactuando con el programa.



OPCIÓN "W"

Ninguna modificación tiene validez hasta que no se usa la opción "w" para grabar la nueva tabla de partición.

m: con la sola pulsación de esta tecla, se desplegará un listado con todos los comandos.

p: nos da un listado de las particiones.

n: crear una nueva partición. Las particiones se crean por defecto de tipo LINUX Nativo.

t: cambiar el tipo de partición.

1: listar los tipos de partición soportados por fdisk.

q: salir sin grabar la tabla de partición.

w: salir grabando la tabla de partición.

Si bien éstos no son todos los comandos, son los necesarios para que podamos preparar nuestro disco para que albergue LINUX. Vamos ahora paso por paso con el procedimiento. -

1. Usamos **p** para ver el estado de nuestro disco.
2. Identificamos la partición que vamos a usar para LINUX; por lo general, será la que hayamos creado al usar FIPS.
3. Debemos borrarla usando el comando **d**, teniendo cuidado de no borrar la partición DOS/WIN que queremos conservar. Seguramente, la partición Windows será **/dev/hda1**.
4. Usamos nuevamente **p** para ver el resultado y tomar nota de las particiones que quedaron. Usaremos esta información más adelante.
5. Creamos una nueva partición que usaremos como de intercambio "SWAP". Para ello, usamos el comando **n**; cuando nos pregunte si es primaria o extendida, pulsamos **p** para indicar que es primaria. Luego nos pedirá el número de partición; esto dependerá de cuántas particiones conservemos para DOS/WIN. Cuando consultamos la tabla con el comando **p**, vimos el listado de las particiones existentes; la nueva deberá ser correlativa con las anteriores. Acto seguido, se nos consulta sobre el primer cilindro que usará la nueva partición y se nos muestra el rango disponible. Ingresamos el primero. Finalmente, nos pregunta el tamaño de la partición o el último cilindro a usar; en este caso ingresamos "+32M", con lo que le estamos diciendo el tamaño en Megabytes.

6. Como ya dijimos, la nueva partición es del tipo LINUX Nativo y, como será la de intercambio, la modificamos usando el comando **t** e ingresando el tipo "82" (LINUX Swap).
7. Usamos nuevamente **p** para ver nuestra nueva partición de intercambio.
8. Ahora creamos la partición LINUX Nativo donde estarán los archivos de LINUX. Usamos el comando **n**. Le decimos que será primaria, desde el primer cilindro disponible, hasta el último. En este caso, no es necesario cambiar el tipo de partición, porque como ya dijimos, por defecto se crean particiones del tipo LINUX Nativo.
9. Finalmente, verificamos nuestra tabla usando **p**.
10. Ésta es la última oportunidad para arrepentirse; de ser así, abortamos todo con la opción **q**.
11. Si todo está bien, como es de esperar, ahora podemos usar el comando "**w**" para escribir la nueva tabla de partición.
12. Pulsamos sobre **done** y tal vez nos informe que es necesario reiniciar la máquina; esto ocurre si modificamos las particiones extendidas. Luego de dar **OK**, la máquina se reiniciará y deberemos seguir los mismos pasos hasta llegar al punto de partición de Disco; entonces, vamos directamente a **done**.

Formateo de particiones

Un cuadro de diálogo nos muestra nuestra nueva partición LINUX Nativa para seleccionarla; en caso de tener más de una, nos mostrará todas. Aquí debemos escoger cuál será la partición que se montará en la raíz del sistema de archivos **/**. Ésta debe ser obligatoriamente del tipo LINUX Nativo. Seleccionamos y pulsamos Enter. (**Figura 7**).

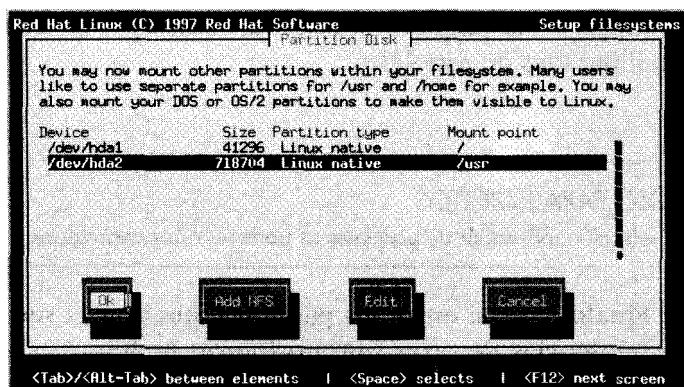


Figura 7. Al seleccionar los paquetes, debe considerarse el uso que se le dará a la PC.

Aquí nos encontramos con una perlita de LINUX: nos muestra las particiones no LINUX, es decir que, si tenemos una partición DOS/WIN, podremos montarla a nuestro sistema de archivos para que sea visible desde LINUX. Normalmente usaremos subdirectorios con nombre como `/dosc` para referirnos al disco C de DOS o `/wind` en el caso de que sea el disco D de Windows; no es éste el caso de las lectoras de CDs, que veremos más adelante.

Seleccionaremos la partición SWAP o de intercambio, activando la opción de chequear sectores defectuosos.

Estamos ahora ante un nuevo cuadro que nos pregunta qué partición formatear. Es conveniente que activemos el testeo de sectores defectuosos, porque si bien el formateo va a demorar un poco más, siempre es bueno estar seguros. La partición SWAP tiene formato especial de tipo 82.

Selección de paquetes

En este punto es donde vamos a tardar más, porque Red Hat pone a nuestra disposición una completísima colección de software para ser instalado, organizado en paquetes que se denominan RPM. Estos paquetes, a su vez, están cuidadosamente catalogados y cuentan con una breve aunque concisa descripción de cada uno.

La ventana de la **Figura 7** nos muestra una discriminación de paquetes en grupos específicos.

Los usuarios con algo de experiencia pueden activar la opción de seleccionar paquetes individuales. Nosotros no lo haremos así, porque la cantidad de paquetes que tendríamos que comentar excedería las posibilidades de este libro.



HAY QUE SABERLO

SÓLO PARA EXPERTOS

La selección individual de paquetes es para usuarios avanzados.

Finalmente, si entre los paquetes que hemos seleccionado existen dependencias de otros que no fueron incluidos, el instalador nos lo advierte dándonos la opción de añadir estos paquetes. Veamos un ejemplo en la **Figura 8**.

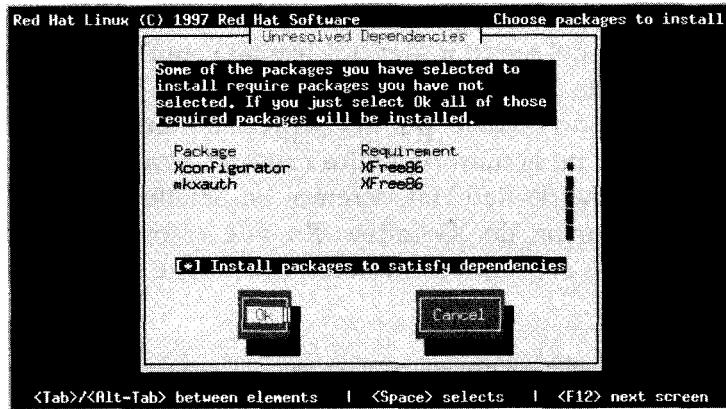


Figura 8. Los paquetes faltantes se instalan automáticamente.

Ahora el instalador dará la orden de formatear las particiones seleccionadas previamente, lo que puede demorar unos minutos dependiendo del tamaño que tengan y de si se ha activado la opción de verificar sectores defectuosos.

Finalmente, se inicia la descompresión y copiado de los paquetes al disco rígido, lo cual también tomará unos minutos.

Terminamos con la instalación de los archivos.

Configuración

Mouse

LINUX intenta identificar el tipo de mouse y el puerto al que está conectado. Una vez que lo ha logrado, nos preguntará el protocolo del mouse (normalmente, Microsoft o PS/2 en los Pentium II) y el número de botones.

XWindow

Este paso no siempre tiene un resultado feliz, pero no hay que preocuparse en caso de falla; luego veremos cómo corregir los posibles problemas.

Si contamos con una distribución comercial de Red Hat, disponemos del servidor Metro-X de la empresa MetroLink. En caso de seleccionar

esta opción, el programa de instalación correrá un programa de configuración del servidor Metro-X llamado **configX**. Este programa es muy sencillo de usar y muy completo, por lo que la configuración mediante este método no tiene por qué representar un problema.

Como en la mayoría de los casos no estaremos instalando una versión oficial de Red Hat, veremos en detalle el método tradicional de configuración de XWindow. En este caso, el servidor gráfico es el **XFree86** y el programa de configuración de Red Hat es **Xconfigurator**.



MÁS DATOS

PLACAS DE VIDEO

Si nuestra placa de video no se encuentra entre las soportadas, podemos tratar con el servidor genérico SVGA.

El programa primero intentará reconocer nuestra placa de video. Lo más probable es que lo logre sin problemas, pero si no sucede así, nos presentará una extensa lista de placas de video de la que escogeremos la nuestra.

Si nuestra placa de video no se encuentra en esta lista, aún nos queda la opción de seleccionar **unlisted card**, donde deberemos suministrar al programa la información técnica de nuestra placa de video. Toda esta información se encuentra normalmente en los manuales o en los documentos que vienen incluidos en los disquetes de drivers para DOS/WIN.

Una vez terminado este paso, el programa de instalación copia el servidor adecuado a nuestro disco. Luego nos presenta la configuración del monitor. Para esto, también nos da una lista con todos los monitores soportados; seleccionamos el adecuado. Si nuestro monitor no está incluido en la lista, la instalación nos permite seleccionar la opción **custom** para luego pasar los parámetros adecuados.



HAY QUE SABERLO

COMIENZO DE LA INSTALACIÓN

No olvidemos que, antes de todo esto, se detalló la información que debíamos recabar para comenzar la instalación; éste es el momento de usarla.

En este paso, debemos informar sobre la cantidad de memoria de video disponible.

Luego nos encontramos con la configuración del reloj del chipset de nuestra placa de video. Lo aconsejable es seleccionar la opción por defecto, a menos que tengamos los conocimientos necesarios para optar por una frecuencia específica.

Sólo nos resta configurar las resoluciones que queremos usar. Para ello contamos con una lista dividida en tres columnas que corresponden a 8, 16 y **24** bits de profundidad por pixel, es decir, 256, 65535 y 16 millones de colores. Usamos las flechas para movernos por las resoluciones y **tab** para cambiar de columna. Para seleccionar una, usamos la barra espaciadora.

Red

El protocolo utilizado por defecto por todos los UNIX es TCP/IP. Aquí usaremos los datos que nos suministró nuestro administrador del sistema. Sólo debemos introducir los valores adecuados; en caso de no estar conectados a una red, obviaremos este paso.

Huso Horario

En el momento de configurar el huso horario, LINUX nos presenta un extenso listado con los husos de todo el mundo. Como podemos ver en la **Figura 9**, sólo debemos buscar la opción adecuada. En mi caso, que vivo en Jujuy, existe "América:Jujuy", así que sólo es cuestión de encontrar nuestra ciudad si es capital de provincia. Si no se encuentra en la lista, usaremos "América: Buenos Aires", puesto que para todo nuestro territorio nacional rige el mismo huso horario.

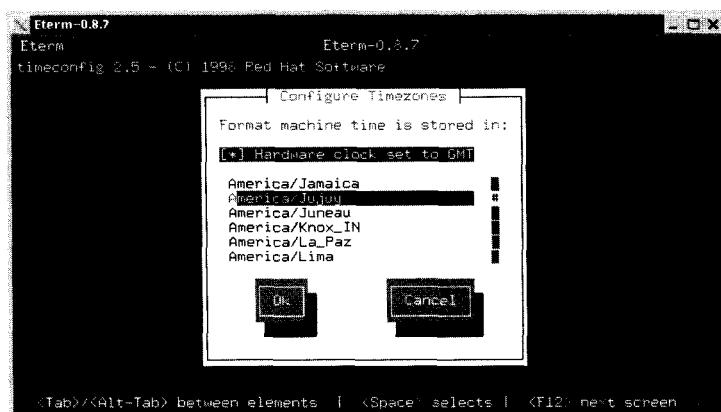


Figura 9.

Servicios



HAY QUE SABERLO

SERVICIOS EN EL BOOTE

Los servicios a lanzar en el booteo dependerán del uso que le demos a la PC.

En LINUX, tenemos los demonios, que no son más que programas que se lanzan para correr en background y que se encargan de atender distintos servicios que nuestro sistema brindará a sus usuarios. Por ejemplo, el demonio lpd se encarga de administrar el recurso de la impresora entre todos los usuarios, controlando, por ejemplo, la cola de impresión. Si no conocemos qué hace cada uno de ellos, es aconsejable que dejemos la lista como está, ya que la instalación sugiere por defecto todos los demonios necesarios de acuerdo con los paquetes seleccionados.



MÁS DATOS

IMPORTANTE

La selección de demonios a lanzar debe basarse en las necesidades.

Impresora

El sistema nos permite configurar el uso de una impresora, brindándonos tres opciones. La impresora puede ser local, es decir que está conectada directamente a nuestra PC. Puede ser una impresora remota, conectada a la red, por lo que necesitaremos la ayuda del administrador de la red para configurarla.



MÁS DATOS

IMPRESORA REMOTA

Consultar los datos con el administrador del sistema si la impresora es remota.

Finalmente la impresora puede estar conectada a otra PC, a través de una LAN (Red de Área Local) o Samba (protocolo que nos permite conectar LINUX con una Red Windows 95 o NT).

Vamos a suponer que nuestra impresora se encuentra conectada directamente a nuestra PC, es decir que se trata de una impresora local.

Un cuadro nos presenta el nombre del recurso y el directorio de

spool. Luego, la instalación intenta detectar el puerto donde está conectada nuestra impresora. Una vez seleccionada, continuamos pulsando sobre el botón **next**, lo que nos lleva a una serie de pantallas donde deberemos especificar cuál es la marca y el modelo de nuestra impresora, y la resolución que usaremos para imprimir, además del tamaño de papel y otras opciones propias de cada equipo.

Clave de root

La instalación nos solicita que definamos la clave de root, es decir, la clave que usaremos para ingresar a nuestro sistema. Es importante tomarnos un tiempo en seleccionar adecuadamente la clave, porque nuestra privacidad dependerá de ella. Puede suceder que la PC en la que estamos instalando LINUX sea la de nuestra casa, por lo que posiblemente no necesitemos mucha seguridad, o por el contrario, puede ser que la PC sea la de nuestro trabajo y necesitemos preservar mejor nuestra información. A continuación, una serie de consejos para que la clave que seleccionemos sea lo suficientemente segura.

- Debe tener, por lo menos, ocho caracteres de largo.
- No usar palabras existentes ni nombres (por ejemplo, Elisa).
- No usar fechas ni números relacionados con el usuario (por ejemplo, el número de DNI).
- No usar secuencias lógicas (por ejemplo, a1b2c3d4).
- No escribir la clave en ningún lado; sólo guardarla en nuestra memoria.
- Una buena clave es aquella que incluye letras mayúsculas y minúsculas, y números al azar (por ejemplo, dF45A2d4).

Deberemos ingresar dos veces la misma clave para su verificación.

LILO

LILO es el programa que nos permite bootear desde distintas particiones o discos con diferentes Sistemas Operativos.



QUITAR LILO

Si quiere quitar LILO de su masterboot, puede hacerlo desde DOS usando fdisk /mbr.

MÁS DATOS

Tenemos dos opciones: la primera (instalar LILO en el Master Boot Record MBR) la usaremos si no tenemos instalado ningún otro programa Boot Manager, tal como System Commander o el de OS/2; la segunda la utilizaremos en caso de que ya estemos usando alguno de estos programas. Si no instalamos LILO, nuestra PC arrancará con el S.O. que estaba instalado antes de LINUX, y para arrancar deberá usar un disco de arranque LINUX, así que es importante que lo instalemos.



MÁS DATOS

ARRANQUE

LILO arranca LINUX por **defecto**.

Reiniciando el sistema

Listo, la instalación está terminada; ahora sólo nos resta prepararnos para reiniciar nuestra PC. A continuación, los pasos necesarios.

- 1) Estamos en una pantalla que nos dice que la instalación de LILO ha finalizado con éxito y nos invita a resetear nuestra maquina. Lo hacemos.
- 2) Quitamos cualquier disquete que se encuentre en nuestro drive.
- 3) Cuando en nuestra pantalla aparece LILO:, pulsamos **tab** para que nos muestre las opciones con las que contamos. Seleccionamos LINUX.
- 4) Luego del arranque, estamos en la pantalla de ingreso al sistema. Nos preguntará:
Login: donde pondremos **root**.
Password: es la clave que cargamos durante la configuración.
Luego de ingresarla, nos presenta el prompt del shell, en este caso, bash.

USUARIOS

| | |
|--------------------------|---------|
| Conceptos | Pág. 51 |
| El archivo PASSWD | Pág. 52 |
| Agregar un usuario | Pág. 54 |
| Borrar un usuario | Pág. 54 |
| Grupos | Pág. 55 |

“...Hay capacidades que son, desde luego, características de la inteligencia:

- Responder muy *flexiblemente* a las situaciones.
- Sacar provecho de circunstancias fortuitas.
- Hallar sentido en mensajes ambiguos o contradictorios.
- Reconocer la importancia relativa de los diferentes elementos de una situación.
- Encontrar semejanzas entre varias situaciones, pese a las diferencias que puedan separarlas.
- Descubrir diferencias entre varias situaciones, pese a las semejanzas que puedan vincularlas.
- Sintetizar nuevos conceptos sobre la base de conceptos viejos que se toman y se reacomodan de nuevas maneras.
- Salir con ideas novedosas.

Todo esto, según Hofstadter, puede enseñársele a una computadora. Ya sería más que bueno enseñárselo a cualquier político...”

La letra e, Augusto Monterroso

Conceptos

Veremos aquí algunos conceptos que serán necesarios para avanzar en el siguiente capítulo. Sólo los trataremos en forma general, ya que más adelante analizaremos estos temas con mayor detalle.



HAY QUE SABERLO

REGISTRO DE TODOS LOS USUARIOS

En el archivo **passwd** del directorio /etc están registrados todos los usuarios.

Denominamos Usuario a toda entidad física o no, que esté registrada en el sistema, el cual tiene asignados recursos para el aprovechamiento del equipo. Si no se es usuario, no se puede usar el sistema.

El único que puede registrar usuarios es el superusuario o root. Se trata del administrador del sistema, es decir, del usuario con mayor poder y privilegios. Es único en cada sistema y es quien dicta las políticas de uso y control del equipo.

A su vez, todos los usuarios se organizan en grupos. Esto significa que un usuario que se dedica a la programación y a la publicidad en una empresa donde se encuentra nuestro hipotético sistema pertenecerá a los

grupos de programación y publicidad (creados por el root). Allí se encontrará con personas con sus mismos intereses y obligaciones, de modo que podrán compartir archivos entre todos.

Gran parte de la seguridad del sistema depende de la correcta administración y control de los usuarios. A la hora de planificar políticas de usuarios, el administrador puede optar por una excesiva seguridad, hecho que resultará molesto para los usuarios, o por ser muy confiado, actitud que seguramente traerá complicaciones con los usuarios experimentados y los hackers. La opción más adecuada es, como siempre, una posición intermedia. Administrar un sistema es algo así como cuidar a la novia o a la esposa: si somos muy celosos, corremos el riesgo de volvemos insoportables; si mostramos desinterés, seguramente la perderemos.

Independientemente de que haya muchos usuarios o no en nuestro sistema, es importante comprender los aspectos de la gestión de usuarios bajo LINUX. Aun si somos el único usuario, debemos tener una cuenta distinta de root para hacer la mayor parte del trabajo. Cada persona que utilice el sistema debe tener su propia cuenta. Es una muy mala idea que varias personas compartan la misma; no sólo es un problema de seguridad, sino que las cuentas se utilizan para identificar unívocamente a los usuarios del sistema y saber quién está haciendo cada cosa en un momento determinado.

archivo PASSWD

MÁS DATOS

SEGURIDAD

Las claves de los usuarios se encuentran encriptadas para mayor seguridad.

Para la correcta administración de los usuarios, tendremos que cargar toda la información que el sistema necesita a fin de identificar a cada uno de ellos. La mayor parte de esta información está almacenada en el archivo **passwd**, que se encuentra en el directorio **/etc**.

Veamos un detalle de esa información:

- Nombre de usuario: es el identificador único dado a cada usuario del sistema. Ejemplos de nombres de usuario son *juan*, *jerez* y *mariag*. Se

pueden utilizar letras y dígitos junto a los caracteres “_” (subrayado) y “.” (punto). Los nombres de usuario se limitan normalmente a ocho caracteres de longitud.

- User ID o UID: es un número único dado a cada usuario del sistema. El sistema normalmente mantiene la pista de la información por UID, no por nombre de usuario.
- Group ID o GID: es la identificación del grupo del usuario por defecto. Cada usuario pertenece a uno o más grupos definidos por el administrador del sistema.
- Clave: el sistema también almacena la clave encriptada del usuario. El comando passwd se utiliza para poner y cambiar las claves de los usuarios.
- Nombre completo o real: se almacena junto con el nombre de usuario. Por ejemplo, el usuario jperez se llama "Juan Pérez" en la vida real.
- Directorio inicial: es el directorio en el que se coloca inicialmente al usuario en tiempo de conexión. Cada usuario debe tener su propio directorio inicial, normalmente situado bajo /home.
- Shell de inicio: es el intérprete de comandos que arranca para el usuario en tiempo de conexión, por ejemplo, /bin/bash y /bin/tcsh.

Cada línea del archivo passwd contiene información acerca de un único usuario; el formato de cada línea es:

nombre:clave encriptada:UID:GID:nombre completo:dir.inicio:shell

Un ejemplo puede ser:

jperez:Xv8Q981g71oKK:102:100:Juan Perez:/home/jperez:/bin/bash

Como se puede ver, el primer campo, "jperez", es el nombre de usuario. El siguiente, "Xv8Q981g71oKK", es la clave encriptada. Las claves no se almacenan en el sistema en ningún formato legible por el hombre. Las claves se encriptan utilizándose a sí mismas como claves de encriptación. En otras palabras, sólo si se conoce la clave se la puede desencriptar. Esta forma de encriptación es bastante segura. Algunos sistemas utilizan "claves en sombra" o *shadow password*, en las que la información de las claves se relega al fichero /etc/shadow. El tercer campo, 102, es el UID, que debe ser único para cada usuario. El cuarto campo, 100, es el GID, que indica que este usuario pertenece al grupo numerado 100. La información de grupos, como la información de

usuarios, se almacena en el fichero `/etc/group`. El quinto campo es el nombre completo del usuario, Juan Pérez. Los dos últimos campos son el directorio inicial del usuario (`/home/jperez`) y el shell (`/bin/bash`), respectivamente.



MÁS DATOS

DATOS ÚTILES

No es necesario que el directorio inicial de un usuario tenga el mismo nombre que el del nombre de usuario. Sin embargo, esto ayuda a identificar el directorio.

Agregar un usuario

Cuando se añade un usuario, se deben seguir varios pasos. Primero, se le debe crear una entrada en `/etc/passwd`, con un nombre de usuario y UID únicos, y se especifica el GID, el nombre completo y el resto de la información. También hay que crear el directorio inicial, poner los atributos en el directorio para que el usuario sea el dueño, suministrar ficheros de comandos de inicialización en el nuevo directorio y hacer alguna otra configuración del sistema (por ejemplo, preparar un buzón para el correo electrónico entrante del nuevo usuario). Aunque todo esto puede hacerse a mano, es mucho más sencillo y seguro usar un programa que se provee con LINUX llamado **adduser**. Esta utilidad nos irá preguntando uno a uno los datos necesarios y, una vez completados, se encargará de modificar todos los archivos de configuración y de crear los directorios que correspondan.

Borrar un usuario

Para borrar un usuario, nos basta con borrarlo del archivo `passwd` y eliminar también todos sus archivos y directorios. Esto se puede hacer a mano o usando **deluser**, que es más seguro. También podemos deshabilitar al usuario colocando un asterisco en el lugar donde va la clave encriptada. Aquí es importante saber que el superusuario puede cambiar en cualquier momento la clave de un usuario, mientras que cada usuario sólo puede cambiar la suya. Así también se puede habilitar a los usuarios para que, por ejemplo, cambien su shell y puedan elegir el que más les guste.



MÁS DATOS

ELIMINAR UN USUARIO

Al eliminar un usuario, también se elimina su directorio home y todo su contenido.

Grupos

Por ser LINUX un sistema multiusuario, necesitamos un método para reunir personas con intereses o tareas comunes; para esto usamos el concepto de Grupo. Un grupo no es más que una definición y sus integrantes. Así, cada usuario pertenece a uno o más grupos. La única importancia real de las relaciones de grupo es la perteneciente a los permisos de ficheros. Cada fichero tiene un "grupo propietario" y un conjunto de permisos de grupo que definen de qué forma pueden acceder al fichero los usuarios del grupo.

Hay varios grupos definidos en el sistema, como **bin**, **mail** y **sys**. Los usuarios no deben pertenecer a ninguno de ellos, sino que se utilizan para permisos de ficheros del sistema. En su lugar, los usuarios deben pertenecer a un grupo individual, como **users**. También se pueden mantener varios grupos de usuarios, como estudiantes, directivos, profesores, soporte y facultad.



HAY QUE SABERLO

GID

Recuérdese que en /etc/passwd, cada usuario tiene un GID por defecto.

El fichero **/etc/group** contiene información acerca de los grupos. El formato de cada línea es:

nombre de grupo:clave:GID:otros miembros

Algunos ejemplos de grupos pueden ser:

- **root**:*:0:
- **usuarios**:*:100:jperez, lmamani
- **invitados**:*:200:
- **otros**:*:250:alumno

El primer grupo, root, es un grupo especial del sistema reservado para la cuenta root.

[MÁS DATOS](#)

PERTENECER A MÁS DE UN GRUPO

Los usuarios pueden pertenecer a más de un grupo, añadiendo sus nombres de usuario a otras líneas de grupo en **/etc/group**.

El siguiente grupo de la lista, **users**, es para usuarios normales. Tiene un GID de 100. Los usuarios jperez y lmamani tienen acceso a este grupo. El tercer grupo, **invitados**, es para usuarios invitados. Finalmente, **otros** es para otros usuarios. El usuario alumno tiene acceso a este grupo.

El comando **groups** lista a qué grupos se tiene acceso. Como se puede ver, el campo **clave** de **/etc/group** raramente se utiliza. A veces, se lo emplea para dar una clave para acceder a un grupo, aunque pocas veces es necesario. Para evitar que los usuarios cambien a grupos privilegiados (con el comando **newgroup**), se pone el campo de la clave en “*”. Se pueden usar los comandos **addgroup** o **groupadd** para añadir grupos a su sistema, aunque también se pueden añadir líneas a **/etc/group** directamente, puesto que no se necesitan más configuraciones para crear un grupo. Para borrar un grupo, sólo hay que borrar su entrada de **/etc/group**.

COMPONENTES

| | |
|------------------------------|---------|
| Cuentas | Pág. 59 |
| Sesión | Pág.59 |
| Terminal | Pág.59 |
| Arranque de LINUX | Pág. 60 |
| init y su descendencia | Pág. 60 |
| Login y password | Pág. 61 |
| Entrada y salida | Pág. 61 |
| Terminales virtuales | Pág. 62 |
| Shell | Pág.62 |
| Variables de entorno | Pág. 64 |
| Prompt | Pág. 64 |
| Midnight Commander | Pág. 65 |
| Aplicaciones | Pág. 66 |

"Trurl y Clapaucio eran alumnos del gran Cerebron Emtadrata, quien durante cuarenta y siete años, había enseñado en la Escuela Superior de Neantica la Teoría General de Dragones. Como sabemos, los dragones no existen. Esta constatación simplista es, tal vez, suficiente para una mentalidad primaria, pero no lo es para la ciencia. La escuela superior de Neantica no se ocupa de lo que existe; la banalidad de la existencia ha sido probada hace demasiados años. Así pues, el genial Cerebron atacó el problema con métodos exactos descubriendo tres clases de dragones: los iguales a cero, los imaginarios y los negativos."

Ciberiada, Stanislaw Lem

Cuentas

Al estar registrado, cada usuario es dueño de una cuenta en el sistema. Esto quiere decir que se le asignan recursos tales como espacio de disco, directorio propio, pertenencia a grupos del sistema, aplicaciones, impresoras, etc. Para ser poseedor de una cuenta, el sistema debe contar con cierta información sobre nosotros.

Sesión

Una vez que somos usuarios y tenemos una cuenta, podemos iniciar una sesión de trabajo.

Una sesión se puede definir como el período de tiempo entre el momento en que nos registramos como usuarios en uso del sistema y el momento en que terminamos de usarlo. Durante este lapso, el sistema operativo administra nuestros pedidos y nos permite hacer uso de los recursos que nos fueron asignados.

Terminal

La terminal es la combinación, como mínimo, de un dispositivo de entrada (teclado) y uno de salida (monitor) conectados al sistema y en interacción con él. Existen dos tipos de terminales:

- Terminales de texto: sólo permiten la visualización por medio de caracteres.

- Terminales gráficos: además del teclado como dispositivo de entrada, tenemos el mouse. En este tipo de terminal, se puede acceder a XWindow, aunque también nos permiten trabajar en modo texto.

Arranque de LINUX

Al encender nuestra computadora, luego de los chequeos de rutina, LINUX lee el masterboot del disco rígido, donde supuestamente tenemos instalado LILO. Luego de que LILO dispone la imagen del kernel desde donde arrancar, procede a cargar el kernel desde el disco rígido. El kernel establece sus tablas y una serie de procesos internos (son los que tienen PID negativo en la tabla de procesos); luego, ejecuta el proceso número uno, que se llama **init**.



MÁS DATOS

Init es el padre de todos los procesos.

init y su descendencia

En LINUX, luego del proceso **init**, se invoca a una tabla llamada **inittab**, que está en el directorio **/etc**; esta tabla contiene una serie de instrucciones para inicializar los periféricos, por ejemplo, los terminales del sistema. En cada terminal activa, **init** arranca un proceso llamado **getty**, que pone el logo del sistema en la terminal y espera el comando de login. El **inittab** puede contener otras directivas para otros dispositivos.

Cuando ha terminado con lo establecido en **inittab**, **init** continúa con llamados a **rc**. Existen varios **rc**, por ejemplo, **rc1**, **rc2**, **rc3**, que están en los directorios llamados **rc1.d**, **rc2.d**, **rc3.d**, respectivamente. **init** corre los comandos de **rc.x**, arrancando para ello un shell. El **rc** que arranquemos en nuestra máquina efectúa comandos necesarios para montar sistemas de archivos, identificar el sistema, configurar la red, arrancar los demonios y, en general, llevar el sistema a un estado donde puede recibir usuarios, es decir, que esté listo para operar.



HAY QUE SABERLO

PROGRAMA "SETUP"

En Red Hat LINUX existe una aplicación que nos permite configurar los procesos que se ejecutarán en el rc, como así también otros procesos de configuración. Se ejecuta a través del programa "Setup".

Cuando un usuario accede a una terminal o consola, causa una interrupción por teclado al iniciar el proceso de login. El programa getty, encargado de atender la terminal, inmediatamente arranca un proceso llamado login; después de esto, el getty muere, pero es reiniiciado cuando el usuario hace el logout, es decir, cuando sale del sistema; init vuelve a arrancar un getty en esa terminal.

Login consulta el archivo **/etc/passwd**, verifica la identidad del usuario y, finalmente, corre el shell del usuario.

Login y password

Cuando iniciemos nuestro sistema por primera vez después de la instalación, éste sólo contará con un usuario autorizado para ingresar: es el superusuario o root.

El login es el proceso de identificación de un usuario para acceder al sistema, y consta de dos pasos. Primero nos requiere nuestro nombre de usuario y luego nuestra clave (password).

Una vez ingresados estos datos y, si éstos son correctos, ya estamos en condiciones de hacer uso del sistema. Lo primero que aparece es la línea de comandos de nuestro shell; allí es donde ingresaremos los comandos y peticiones al sistema.

Para probar, podemos empezar por solicitar un listado del directorio, usando el comando **ls**.

Entrada y salida

Veamos un ejemplo clásico de logueo.



DEFINICIONES

LOGUEO

Se llama así al proceso de registro o ingreso en el sistema.

```
Red Hat LINUX release 5.2 (Apollo)
Kernel 2.2.1 on a i586
amankay login:root
Password:
```

Lo que requiere primero es el nombre de usuario; le decimos **root**. Luego nos pide la clave de ese usuario. Puede suceder que el sistema nos dé el mensaje **login incorrect**, en cuyo caso deberemos intentar una vez más, porque probablemente hayamos cometido algún error de tipeo. Es importante que, antes de abandonar el equipo, cerrremos la sesión iniciada. Para ello, sólo necesitaremos teclear **exit** en la línea de comandos.



HAY QUE SABERLO

Si olvidamos la clave de root, podemos arrancar Linux pasando como parámetro a Lilo la palabra "single"¹ luego de Linux.

Terminales virtuales

Al ser LINUX un sistema multiusuario, contamos con terminales virtuales. Esto significa que, si en nuestro equipo tenemos un solo teclado y un monitor, podemos crear terminales virtuales usando el mismo teclado y display. Por defecto, en LINUX disponemos de seis terminales, a las que accedemos pulsando la combinación de teclas ALT+F1, ALT+F2, ALT+F3, ALT+F4, ALT+F5 y ALT+F6. En cada una de estas terminales, podemos registrar usuarios distintos y ejecutar diferentes aplicaciones o las mismas si queremos.

Desde las XWindow, también se puede acceder a estas terminales virtuales agregando la tecla CONTROL a la combinación correspondiente (por ejemplo, CONTROL+ALT+F2). Para volver a las XWindow, debemos pulsar ALT+F7.

Shell

Una vez que estamos registrados en el sistema, se nos presenta una pantalla con estos datos:

```
Red Hat LINUX release 5.2 (Apollo)
Kernel 2.2.1 on a i586
amankay login:root
Password:

Last login: Wed Sep 16 02:13:00 on ttym1
[root@amankay /root]#
```

Esta última línea es el prompt del shell, la línea de comando que espera nuestras instrucciones para procesarlas. Vamos a analizarla parte por parte.

- **root**: nombre del usuario.
- @ indica pertenencia al siguiente dato.
- **amankay**: hostname, nombre del equipo.
- **/root**: subdirectorio donde nos encontramos.
- #: símbolo del prompt.



MÁS DATOS

Se puede cambiar el prompt del shell modificando la variable `ps1` del archivo `leref profile`.

Un shell es lo que comúnmente se conoce como el intérprete de comandos. Por ejemplo, el shell de DOS es el `command.com`, que nos provee del prompt `C:\>`. Es la última capa de un sistema operativo, la que se encuentra en contacto directo con el usuario. Su principal función es recibir las órdenes del operador, procesarlas y pasarlas a capas inferiores del S.O. para que éste actúe según corresponda.

En LINUX, tenemos una gran variedad de shells. Normalmente, todas las instalaciones instalan por defecto `bash` (*Bourne Again SHell*, Nuevamente el Shell Bourne), aunque se puede cambiar de shell en cualquier momento con la instrucción `chsh`. Bash proporciona una gran variedad de ventajas sobre los otros shell, por lo que resulta ser el favorito. Por ejemplo, tiene capacidad de edición de la línea de comandos para la corrección de errores; posee un historial que nos permite recuperar comandos ingresados; cuenta con capacidad de completar comandos y nombres de archivos; es programable, lo que nos permite crear poderosos scripts de shell; soporta alias y, además, cuenta con un avanzado control de procesos.



PERSONALIZAR EL SHELL

Creando el archivo `.bashrc` en nuestro directorio home, podemos personalizar nuestro shell con alias, funciones, etc.

Variables de entorno

En un shell, tenemos variables de entorno, que se encargan de almacenar información que puede ser requerida y consultada por el S.O. y por las aplicaciones. Por lo general, estas variables almacenan rutas de búsqueda de aplicaciones, nombres de directorios necesarios para algunos programas, números de versiones, etc.

Por ejemplo, si queremos consultar las variables de entorno definidas en nuestro sistema, podemos usar el comando `env`. La variable `PS1` es la que define el path de búsqueda.

Para definir variables de entorno en bash, usamos `export` de la siguiente manera:

```
[root @amankay /root]# export NOMBRE_DE_VARIABLE=valor de la variable
```

Prompt

Ya estamos un tanto cansados de teoría, pero era necesaria para saber dónde estábamos parados. Seguramente estamos todos de acuerdo en que la práctica es el mejor método de aprendizaje, así que hagamos algunos experimentos con las líneas de comandos:

```
# ls -l
total 2341
-m-r-r- 1 root  root  Configuracion-Impresion-Como.txt      69913 Jan 30
17:38
-rw-r-r- 1 root  root  Filtros-Impresion-Como.txt          10782 Jan 30 17:47
-m-r-r- 1 root  root  against.c                           4130 Jan
22 19:11
-m-r-r- 1 root  root  hackpage.zip                         7367 Jan 23 17:33
-m-r-r- 1 root  root  hall1.gif                            11705 Jan 23 18:18
-m-r-r- 1 root  root  hall13.gif                           11705 Jan 23 18:18
-m-r-r- 1 root  root  ircii-current.tar.gz                 600820 Feb
3 00:49
-m-r-r- 1 root  root  killport.c                          4266 Jan
22 19:07
-m-r-r- 1 root  root  lentes
```

```

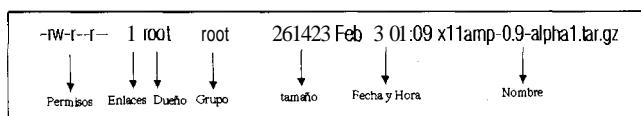
drwxr-xr-x 5 root root libros          1024 Jan 25 17:43
-m-r-r-  1 root root manual.texi      970514 Jan
22 20:29
-m-r-r-  1 root root mikmod-x11amp-0.99.4.tar.gz 190344 Feb  3 01:10
drwx--- 2 root root nsmail          1024 Jan 17 01:44
-m-r-r-  1 root root orcamp.zip       35563 Feb
3 01:12
-m-r-r-  1 root root oshare.c        7138 Jan 26 00:24
-m-r-r-  1 root root oshare_1_gou.c 13687 Jan 27 20:54
-rw-r-r- 1 root root pnp            9777 Feb  3 02:57
-rwxr-xr-x 1 root root pps          102 Feb  1 11:46
-m-r-r-  1 root root pro_diag_scan2.gif 20939 Jan 22 23:27
-rw-r-r- 1 root root resetter21.c   8931 Jan 22 19:08
-rw---  1 root root rev_task        1598 Jul 20 1998
drwx--- 2 root root root           1024 Feb  3 13:10
-m-r-r-  1 root root spoof.c        4480 Jan 27 20:55
-m-r-r-  1 root root sqltut.htm     82896 Jan
22 22:36
-m-r-r-  1 root root uso-impresion.como.txt 21534 Jan
30 17:38
-m-r-r-  1 root root vesa3.0-0.3.tar.gz 12165 Jan 31 16:37
-rw-r-r- 1 root root x11amp-0.9-alpha1.tar.gz 261423 Feb  3 01:09

```

Lo que obtenemos es un listado de los archivos y subdirectorios del directorio donde estamos ubicados. Analicemos una de estas líneas:

```
-m-r-r- 1 root root x11amp-0.9-alpha1.tar.gz 261423 Feb  3 01:09
```

En esta línea tenemos mucha información acerca de un archivo de nuestro directorio.



Midnight

Con todas las distribuciones de LINUX se incluye un excelente shell interactivo, el Midnight Commander. En su aspecto general, es muy parecido al Norton Commander de DOS, aunque veremos que es mucho más completo y poderoso.

Como se puede ver en la **Figura 1**, su apariencia es muy tradicional, pero tiene además todas las funciones necesarias para la navegación de nuestros sistemas de archivos y para el manejo de archivos. También presenta ventajas tales como capacidad de conexiones FTP y manejo del servidor como un directorio local.



MÁS DATOS

El autor de MC es un mexicano llamado Miguel de Icaza.

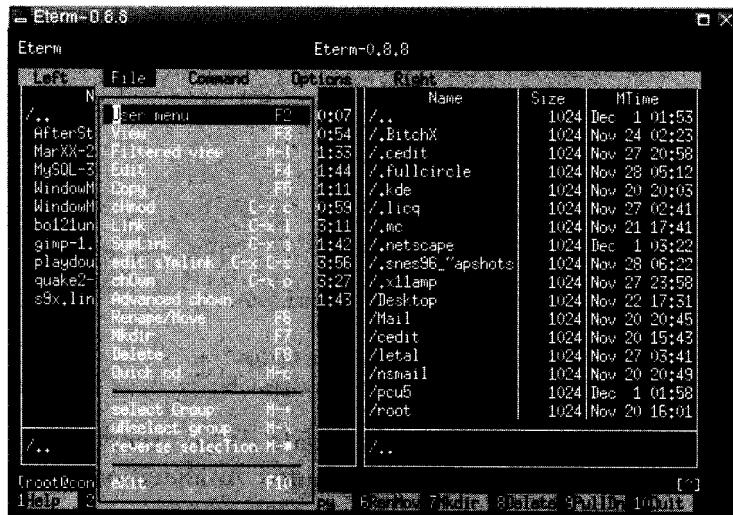


Figura 1. Midnight Commander está desarrollado por un programador mexicano.



MÁS DATOS

MC

Con el tiempo, nos acostumbraremos a usar MC hasta el punto en que, para muchos, será imposible vivir sin él. MC también tiene capacidades de operación en Red, como FTP.

Aplicaciones

En la línea de comandos es donde debemos invocar las aplicaciones o programas que queramos ejecutar. Éstos serán buscados en el path en el orden en que se definieron los directorios que lo componen.

La variable del path puede ser redefinida o ampliada sin problemas desde la línea de comandos o desde el archivo `profile`, cuya función veremos más adelante. Las aplicaciones que no se encuentren en este path no se ejecutarán a menos que las invoquemos con toda la trayectoria desde el punto donde estemos ubicados.

Si la aplicación a ejecutar se encuentra en el directorio actual, veremos que tampoco se ejecuta, puesto que el directorio actual no se incluye en el path. Existen distribuciones que sí lo hacen, como Slackware, e incluso nosotros mismos podemos incluirla, pero no es una buena política.

Para ejecutar las aplicaciones ubicadas en el directorio actual, sólo debemos anteceder al nombre del archivo de la aplicación el path al directorio actual, ya sea describiendo toda la trayectoria o usando en puntero seguido de la barra de directorios /. Así, si lo que queremos es invocar el programa ICQ que se encuentra en el directorio actual, debemos ingresar la sentencia ./ICQ en la línea de comandos.

COMANDOS

| | |
|--------------------------------------|-----------------|
| Formato | .Pág.71 |
| Comandos de archivos | .Pág. 72 |
| Comandos generales | .Pág. 78 |
| Comandos de información | .Pág. 80 |
| Comandos de usuarios | .Pág. 82 |
| Comandos de red | .Pág.82 |

"La voz del intelecto es callada, pero no ceja hasta conquistar una audiencia y, en última instancia, después de interminables repudios, consigue su objetivo. Es éste uno de los pocos aspectos en los que cabe un cierto optimismo sobre el futuro de la humanidad."

... *El futuro de una ilusión, Sigmund Freud*

Formato

El formato que se usa para ingresar comandos desde el prompt del shell es muy sencillo y, por lo general, todas las aplicaciones siguen una regla común.

Veremos que la mayoría de los comandos en LINUX están formados sólo por dos letras de carácter mnemónico. Los comandos reciben parámetros tales como opciones switches u otros parámetros. Por regla general, el orden es: primero las opciones, luego los switches y, finalmente, otros parámetros.

El formato general de los comandos es:

verbo argumento argumento ...

- El verbo, o nombre del comando, es siempre necesario.
- Los argumentos pueden ser necesarios o no, dependiendo del comando u orden.
- Los nombres de órdenes en UNIX son siempre en minúsculas y no pueden ser abreviados.
- Los argumentos pueden ser de dos tipos: opciones o adverbios, precedidos de un guión -, o nombres de archivos.
- LINUX permite el uso de los caracteres comodín ? y *, como así también la descripción de rangos de caracteres.

Por ejemplo, para solicitar un listado de directorio con colores distintos para cada tipo de archivo y con descripción detallada de cada entrada usando caracteres comodín para filtrar los archivos cuyos nombres comiencen con las letras "abc", usaremos la siguiente orden:
ls -l -color=yes abc*.

Todos los comandos tienen su correspondiente página en el manual de LINUX. Conviene consultarlos para ver qué otras opciones se ofrecen.

A continuación, veremos una lista de los comandos más comunes

de este sistema. Están catalogados en cuatro categorías, aunque algunos podrían corresponder a más de una y otros, posiblemente, no se ajusten del todo a la categoría que les correspondió. Esto es así sólo con el fin de que el lector pueda acceder rápidamente a las descripciones del libro.

Comandos de archivos

Comando: cat

Propósito: mostrar el contenido de los archivos especificados.

Sintaxis: **cat [archivo1 archivo2 ...]**

Observaciones:

- Los archivos se muestran hacia la salida estándar en el mismo orden en que se los da en la línea de comandos.
- Si no se especifica ningún archivo, éste se lee de la entrada estándar.

Ejemplos:

- **cat /etc/passwd** (muestra el contenido del archivo **/etc/passwd**)
- **cat ventas presupuesto saldos > informe** (concatena en informes los archivos **ventas**, **presupuesto** y **saldos**)
- **cat >memo** (guarda en **memo** la entrada de teclado, hasta pulsar **CONTROL-D**)

Comando: cd

Propósito: cambiar el directorio de trabajo.

Sintaxis: **cd [directorio]**

Observaciones:

- Si no se especifica el directorio, cambia a HOME.
- Se puede omitir el signo / al final.

Ejemplos:

- **cd notas** (cambia al directorio **notas**)
- **cd /usr/local/public** (cambia al directorio **/usr/local/public**, el signo / al comienzo especifica a la raíz)
- **cd ..** (cambia al directorio **padre**)

Comando: chmod

Propósito: cambiar los permisos de los archivos y/o directorios especificados.

Sintaxis: **chmod [permisos] archivos ...**

Los permisos afectan a:

- u: usuario (dueño)
- g: grupo
- o: otros
- a: todos (usuario, grupo y otros)

Las opciones son:

- +: agregar permisos
- -: quitar permisos
- =: asignar permisos

Los permisos son:

- r: lectura
- w: escritura
- x: ejecución

Ejemplos:

- **chmod +x Bitchx** (da permiso de ejecución a Bitchx, para todos los usuarios)
- **chmod go-r *** (quita permiso de lectura a todos los archivos del directorio actual)
- **chmod g=rw** (fija los permisos para el grupo-en lectura/escritura)

Comando: comm

Propósito: comparar dos archivos.

Sintaxis: **comm [opciones] archivo1 archivo2**

Opciones:

- 1: no lista las líneas que existen sólo en archivo1.
- 2: no lista las líneas que existen sólo en archivo2.
- 3: no lista las líneas que existen en ambos archivos.

Ejemplo:

- **comm carta.old carta** (compara los archivos carta.old y carta)

Comando: cp

Propósito: copiar archivos.

Sintaxis:

```
cp archivo-fuente archivo_destino  
cp archivo1 archivo2 ... directorio_destino (2)
```

Observaciones:

- Se puede cambiar el nombre del archivo al copiar.

- Cuando copia más de un archivo, preserva los nombres en el directorio destino.

Ejemplos:

- **cp dosemu.conf dosemu.conf.bak**
- **cp xinitrc ..** (copia el archivo **xinitrc** en el directorio padre)
- **cp *rpm /backup/** (copia todos los archivos cuyos nombres terminen en **rpm** en el directorio **/backup**)

Comando: diff

Propósito: mostrar las diferencias entre dos archivos.

Sintaxis: **diff archivol archivo2**

Ejemplo:

- **diff carta.back carta** (muestra las diferencias entre los archivos **carta.back** y **carta**)

Comando: file

Propósito: indicar la clasificación de los archivos.

Sintaxis: **file archivol archivo2 ...**

Ejemplo:

- **file ***

Comando: grep

Propósito: mostrar las líneas que cumplan con cierto patrón en los archivos especificados.

Sintaxis: **grep patron archivol archivo2 ...**

Ejemplo:

- **grep contaduria resumen.txt** (muestra las líneas que tengan la palabra **contaduria** de **resumen.txt**)

Comando: ln

Propósito: crear un enlace entre archivos o directorios.

Sintaxis: **ln [-s] archivo-real nombre-enlace**

Opción:

- **-s**: crea un enlace (link) simbólico.

Observaciones:

- Cada vez que se refiere al archivo **nombre_enlace**, se utilizará el archivo **archivo-real**.
- También es válido con directorios.

Ejemplo:

- `ln ~/notas/nota.1 nota.1` (crea un enlace `nota.1` en el directorio actual al archivo `nota.1` que está en `~/notas/nota.1`)

Comando: lpr

Propósito: imprimir archivos.

Sintaxis: `lpr archivol archivo2 ...`

Ejemplo:

- `lpr resument.txt`

Comando: ls

Propósito: mostrar la lista de archivos.

Sintaxis: `ls [opciones] [lista de archivos y/o directorio]`

Opciones:

- `-a`: muestra todos los archivos.
- `-l`: descripción detallada.
- `-t`: muestra los archivos ordenados desde el más reciente al más antiguo.
- `-F`: identifica archivos comunes, directorios y links simbólicos.

Ejemplos:

- `ls`
- `ls -lF notas*`
- `ls -Ft entradas salidas estados *.err`

Comando: mkdir

Propósito: crear un directorio.

Sintaxis: `mkdir directorio`

Ejemplo:

- `mkdir libros` (crea el directorio `libros`)

Comando: more

Propósito: mostrar un archivo por pantalla.

Sintaxis: `more archivol archivo2`

Comandos internos:

- <ESPACIO>: página siguiente
- b: página anterior
- h: ayuda
- q: salir

Ejemplos:

- **more** datos (muestra el archivo datos por pantallas)

Comando: mv

Propósito: mover (o renombrar) archivos.

Sintaxis:

Renombrar: **mv archivo_viejo archivo-nuevo**

Mover: **mv archivo1 archivo2 ... directorio**

Ejemplos:

- **mv memos memos.txt** (cambia el nombre del archivo memos a **memos.txt**)
- **mv *.doc *.txt documentos** (mueve todos los archivos .doc y .txt al directorio documentos)

Comando: pr

Propósito: preparar un archivo para imprimirla.

Sintaxis: **pr [opciones] [archivo1 archivo2 ...]**

Opciones:

- **-h**: coloca el texto que sigue a esta opción al comienzo de cada página
- **-ln**: fija el número de líneas por página en n.
- **-wn**: fija el ancho a n caracteres.

Ejemplo:

- **pr -h 'Informe para el jefe' -166 -w70 informe | lpr**
(prepara e imprime el archivo informe)

Comando: rm

Propósito: borrar archivos.

Sintaxis: **rm [opciones] archivo1 archivo2**

Opciones:

- **-i**: pide confirmación antes de borrar.
- **-r**: borra un directorio y todos los archivos y directorios que están en él.
- **-f**: no pide confirmación antes de borrar.

Ejemplos:

- **rm -i *** (borra todos los archivos del directorio actual pidiendo confirmación)
- **rm *.rpm** (borra todos los archivos terminados en .rpm)
- **rm -rf backup** (borra el directorio backup y todo su contenido sin pedir confirmación)

Comando: rmdir

Propósito: borrar directorios vacíos.

Sintaxis: rmdir **directorio1** **directorio2**

Ejemplo:

- **rmdir** backup (borra el directorio backup; debe estar vacío)

Comando: sort

Propósito: ordenar archivos

Sintaxis: sort [opciones] [campos] [**archivo1** archivo2 ...]

Opciones :

- -b: ignora blanco al principio y al final de la línea.
- -f: considera mayúsculas y minúsculas iguales.
- -n: orden numérico.
- -r: invierte el orden.
- **-tx**: el carácter **x** se considera como separador de campos.

Observaciones:

- La indicación de campos señala bajo qué campos se desea ordenar.
- Si no se indica un archivo de entrada, los datos son leídos de la entrada estándar (teclado).

Ejemplos:

- **who | sort** (muestra la lista de usuarios ordenada alfabéticamente)
- **sort -n notas.alumnos** (muestra las líneas del archivo notas.alumnos ordenadas numéricamente)
- **sort -n +3 padron** (muestra las líneas del archivo padron ordenadas numéricamente según el cuarto campo)

Comando: wc

Propósito: contar las líneas, palabras y caracteres de los archivos.

Sintaxis: wc [opciones] **archivo1** archivo2

Opciones:

- **-c**: cuenta el número de caracteres del archivo.
- **-l**: cuenta el número de líneas del archivo.
- **-w**: cuenta el número de palabras del archivo.

Observaciones:

- Por defecto, cuenta caracteres, líneas y palabras (**-lwc**).
- Si no se especifican archivos, se lee de la entrada estándar (teclado).

Ejemplo:

- **wc *** (cuenta el número de líneas, palabras y caracteres de todos los archivos del directorio actual)

Comando: find

Propósito: buscar nombres de ficheros.

Opciones:

- **name <fichero>**: el fichero a buscar.

Ejemplos:

```
find $HOME -print (lista todos los archivos de $HOME)
find -name "*abc*" -print (lista todos los archivos del sistema con nombre *abc*)
```

Comando: chgrp

Propósito: cambiar el grupo al que pertenece uno o varios archivos.

Sintaxis: **chgrp <opciones> <grupo> <archivos>**

Ejemplo:

```
chgrp alumnos apuntes-historia.txt (cambia el archivo apuntes-historia.txt al grupo alumnos) .
```

Comandos generales

Comando: at

Propósito: ejecutar un comando o un archivo de comandos en la fecha y hora indicados.

Sintaxis: **at hora [dia] [archivo]**

Observaciones:

- Para determinar fecha y hora, siempre se busca el momento más reciente que cumpla las especificaciones.
- Si no se especifica el archivo de comandos, éstos se ingresan desde el teclado terminando con CONTROL-D. Se recomienda utilizar redirecciónamientos para los resultados.

Ejemplos:

- **at 1200**
- **at 2:00pm sunday ejecutar_fsck_/dev/hda1**
- **at 1000 feb 14 enviarcorreo**
- **at 0000 Jan 1 enviar_saludos_año_nuevo**
- **at 1200 Apr 12, 1997 imprimir-informe**
- **at now + 2 hour cerrar-sesion**

Comando: echo

Propósito: mostrar un mensaje en pantalla.

Sintaxis: **echo [-n] mensaje**

Observaciones:

Al especificar la opción -n, la siguiente escritura se hace en la misma línea.

Ejemplos:

- **echo esto es un mensaje** (muestra 'esto es un mensaje' en la pantalla)
- **echo esto lo guardo en un archivo >archivo** (Crea el archivo 'archivo' con el contenido 'esto lo guardo en un archivo')

Comando: expr

Propósito: evaluar una expresión.

Sintaxis: **expr expresión**

Observaciones:

- Cada operando u operador debe estar separado por un espacio del siguiente.

- Operadores :

- +: suma
- -: resta
- *: multiplicación
- /: división

- Para evitar que el carácter * sea interpretado como un carácter comodín, se antepone una \ o se coloca entre comillas; lo mismo sucede con los paréntesis.
- Sólo trabaja con números enteros.
- Las multiplicaciones y divisiones tienen prioridad sobre las sumas y restas.

Ejemplos:

- **expr 2 + 5 (=7)**
- **expr 2 + 7 * 7 (=51)**
- **expr \((2 + 5 \)) * 7 (= 49)**

Comando: kill

Propósito: enviar una señal de fin a un proceso.

Sintaxis: **kill [-sen~al] proceso1 proceso2 ...**

Ejemplos:

- **kill 20192 20194 20201**
- **kill -9 20192 20194 20201**

Comando: man

Propósito: mostrar la página de manual del comando o recurso dado.

Sintaxis: **man -s <Nº de manual><command>**, donde **<command>** es el nombre del comando o recurso sobre el que queremos ayuda.
Observaciones:

- Los manuales estándar de UNIX van numerados del 1 al 7:
 1. Comandos de uso general.
 2. Llamadas al sistema.
 3. Rutinas de las bibliotecas del sistema.
 4. Formatos de los ficheros.
 5. Miscelánea (cabeceras, macros, caracteres, etc.).
 6. Juegos.
 7. Ficheros especiales de dispositivos.
 8. Glosario de términos empleados en los manuales.

Ejemplo:

- **man ls** (muestra ayuda sobre la orden ls)

Comandos de información

Comando: cal

Propósito: mostrar un calendario en pantalla.

Sintaxis: **cal [[mes] año]**

Observaciones:

- El mes debe ser especificado como número y el año como entero de cuatro dígitos.
- Si no se especifican ni el mes ni el año, se muestra un calendario del mes actual.

Ejemplos:

- **cal 2000** (calendario del año 2000)
- **cal 12 1998** (calendario de Diciembre de 1998)

Comando: date

Propósito: mostrar la fecha y hora del sistema.

Comando: \index{finger}finger

Propósito: ver quién está en el sistema y qué hace.

Ejemplo:

- **finger finger -1**

Comando: ps

Propósito: desplegar el estado de los procesos.

Sintaxis: ps [opciones]

Opciones: existen multitud de opciones, algunas de las más comunes son:

- **-e**: información de todo el sistema.
- **-l**: formato largo.
- **-f**: información de los comandos que el sistema está procesando.
- **-aux**: muestra todos los procesos.

Según las opciones que le suministremos al comando, aparecerán diversos campos, entre los que cabe destacar:

- **UID**: número de identificación del usuario.
- **PID**: número de identificación del proceso.
- **PPID**: número de identificación del proceso padre (aquel proceso que creó a éste).
- **PRI**: prioridad (cuanto más alto sea este número, menor es la prioridad).
- **NI**: número "nice" (es la prioridad efectiva; sólo se puede subir o, lo que es lo mismo, bajar la prioridad).
- **TIME**: tiempo de CPU en min:sg.
- **COMD**: nombre del comando que se está ejecutando.

Ejemplos:

- **ps -1** (muestra una lista detallada de los procesos)
- **ps -aux** (muestra todos los procesos, incluso aquellos que no tienen una terminal asignada)

Comando: tty

Propósito: devolver el nombre del terminal que estamos usando.

Comando: \index{stty}stty

Propósito: mostrar la configuración de la terminal. Permite cambiar los parámetros de ésta.

Comando: pwd

Propósito: devolver el directorio actual.

Ejemplo: **pwd**

Comando: df

Propósito: mostrar información de bloques libres en los discos montados.

Comandos de usuarios

Comando: adduser

Propósito: agregar un usuario al sistema.

Ejemplo:

- **adduser jperez** (crea una cuenta de usuario para **jperez**)

Comando: userdel o deluser

Propósito: borrar un usuario del sistema.

Ejemplo:

- **userdel jperez** (borra la cuenta de **jperez** con todos sus directorios y archivos)

Comando: passwd

Propósito: cambiar la clave de un usuario

Ejemplo:

- **passwd jperez** (nos pide la nueva clave para el usuario **jperez**; si usamos **passwd** sin parámetros, cambiamos la clave del usuario en uso)

Comandos de red

Comando: mail

Propósito: enviar o recibir correo.

Sintaxis: **mail [usuariol usuario2 ...]**

Observaciones:

- Al especificar una lista de usuarios, envía la carta a los usuarios especificados.

- 1 Para terminar una carta, se coloca un . (punto) o se pulsa CONTROL D en la primera columna.
- Al dar el comando mail sin parámetros, se ingresa al modo de comandos internos de mail.

Comando: mesg

Propósito: permitir o prohibir la recepción de mensajes.

Sintaxis: **mesg** y (permite) **mesg n** (prohibe)

Comando: who

Propósito: mostrar la lista de usuarios conectados al sistema.

Sintaxis: **who [am i]**

Observaciones:

- 1 Al dar los parámetros **am i**, muestra la información de la conexión en curso; en caso contrario, muestra la lista de todos los usuarios conectados.

Ejemplo:

- 1 **who** (muestra los usuarios conectados)

Comando: write

Propósito: enviar un mensaje a otro usuario conectado al sistema.

Sintaxis: **write usuario [tty]**

Observaciones:

- 1 El mensaje es leído desde el teclado; la conversación finaliza con CONTROL D.
- 1 **tty** se refiere a la terminal con la cual quiere establecer la comunicación, en caso de que el usuario esté en más de una terminal.
- 1 Sólo es posible enviar un mensaje si el destinatario lo permite.

Ejemplos:

Usuarios jperez y jlopez

- **write jlopez**

Message from jperez on tty02 at 12:27 ...

- **write jperez tty02**

**Message from agonzale on tty13 at 12:28 Hola esto es un
mensaje**

Hola muy lindo tu mensaje(CTRL-D)

SISTEMA DE ARCHIVOS

| | |
|----------------------------------|---------|
| Universo LINUX | Pág. 87 |
| I-nodo | Pág. 87 |
| Sistemas montados | Pág. 88 |
| Estructura | Pág. 89 |
| Permisos | Pág. 91 |
| Interpretación de permisos | Pág. 93 |
| Dependencias de permisos | Pág. 93 |
| Cambiando permisos | Pág. 94 |

"Por regla general, las sociedades humanas no son innovadoras, sino más bien jerárquicas y ritualistas. Cualquier sugerencia de cambio se acoge con recelo, ya que implica la incómoda transformación futura del ritual y la jerarquía imperante... Sin embargo, llega el momento en que es preciso que las sociedades cambien."

Los Dragones del Edén, Carl Sagan

Universo LINUX

El ambiente LINUX es un sistema de archivos, es decir que todo lo que compone el sistema está representado por un archivo. Incluso cada uno de los dispositivos, periféricos y procesos en ejecución son ficheros.

Obviamente, todos estos archivos permiten su lectura y escritura, lo que redunda en una simplicidad para los programadores por la统一化 de las entradas y salidas.

I-nodo

En LINUX, cada archivo se correlaciona únicamente con una estructura de control llamada i-nodo. La fila de bloques, pues, se divide tal como se observa en la Figura 1.

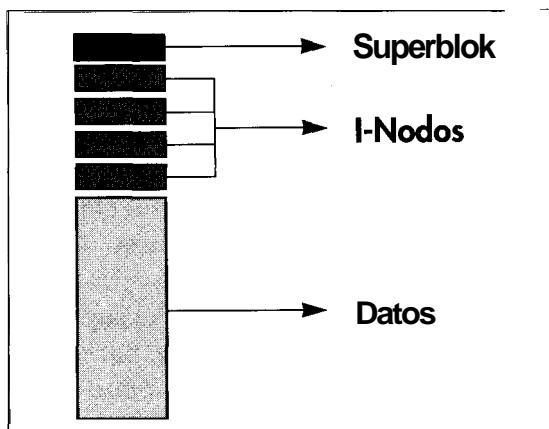


Figura 1. División de la fila de bloques, con los siguientes elementos: el superblok, los i-nodos y los bloques de datos.

El superbloque tiene la información principal sobre la estructura del sistema de archivos. Los i-nodos contienen la información fundamental acerca de cada archivo, exceptuando el nombre de éstos. El área de datos tiene el archivo en sí, es decir, su contenido.

Sistemas montados

En LINUX estamos totalmente abstraídos del concepto de unidades, lo cual permite que un sistema de archivos forme parte integral de la estructura de otro. Esto sirve para obviar todo concepto de unidad desde el punto de vista del usuario, como dijimos anteriormente.

De esta manera, para tener acceso a dispositivos externos de almacenamiento, por ejemplo disquetes, zip-disc u otros disco rígidos, primero debemos incorporar este dispositivo a nuestro sistema de archivos. Esto se logra con la instrucción **mount**, que se analiza en detalle en el **Capítulo 6, Comandos**. Por ejemplo, supongamos que queremos tener acceso a nuestra lectora de CD-ROMs, que se encuentra conectada a la segunda controladora IDE como maestro, es decir que correspondería al dispositivo **/dev/hdc** en LINUX. Obviamente, el formato de este disco compacto de datos es **iso9660** y lo queremos montar en el directorio creado a tal efecto, llamado **/cdrom**. El comando a utilizar sería:

```
# mount -t iso9660 /dev/hdc /cdrom -r
```

Analicemos esta línea parte por parte. Primero tenemos el comando o verbo **mount**; luego, con el switch **-t** le especificamos que la cadena que sigue, **iso9660**, es el sistema de archivos que vamos a montar. El siguiente parámetro le dice al comando que el device driver del dispositivo es **/dev/hdc**. Luego le decimos el punto de nuestro sistema de archivos donde vamos a montar el nuevo dispositivo, y finalmente, mediante **-r**, especificamos que el dispositivo es de sólo lectura. Para que esto funcione, tanto el i-nodo como el superbloque de los sistemas de archivos contienen información que hace posible el funcionamiento de **mount**; por ejemplo el i-nodo contiene espacio para este propósito.



HAY QUE SABERLO

No siempre es necesario especificar el formato del sistema de archivos. Las últimas versiones del **Kernel** lo reconocen automáticamente.



HAY QUE SABERLO

Por medio de **NFS**, es posible montar un sistema de archivos aunque éste resida en una máquina remota.

Estructura

Al igual que en DOS, en LINUX tenemos un sistema de archivos jerárquico de directorios-ficheros.

No existe el concepto de volumen, es decir que todos los discos y particiones pertenecen a una sola estructura que cuelga de un directorio llamado raíz y que se identifica con **/**. Un archivo es un conjunto de información encapsulada en una sola entidad.

Existen cuatro tipos de ficheros en LINUX:

1. Ordinarios: son cadenas de bytes terminadas con un carácter de fin de archivo **<ctrl>D**.
2. Directorios: contienen nombres de ficheros y sus direcciones físicas.
3. Especiales: asociados a dispositivos entrada/salida.
4. Virtuales: son aquellos que representan procesos en ejecución o información del sistema.

Existen además tres ficheros estándar implementados en UNIX:

1. Entrada estándar (**stdin**): Teclado (0)
2. Salida estándar (**stdout**): Pantalla (1)
3. Errores estándar (**stderr**): Pantalla (2)

Un sistema de ficheros tiene una estructura de árbol y parte de un lugar especial llamado **root** (raíz), que se simboliza por **/**.

Todos los usuarios tienen asignado un subdirectorío de trabajo llamado **HOME**. En el caso del superusuario, será el subdirectorío **/root**.

Una ventaja de LINUX es su soporte para varios sistemas de archivos, entre los que se pueden destacar MSDOS, VFAT, System V, Minix y OS/2.

En cada subdirectorío, tenemos dos entradas: **..** y **..**, que son punteros al subdirectorío actual y al subdirectorío padre, respectivamente.



MÁS DATOS

Con **Vfat**, podemos acceder en forma transparente a discos con formato fat 32.



HAY QUE SABERLO

MUY IMPORTANTE

Cuando no tenemos referenciado el subdirectorio actual en la variable **PATH**, tendremos que ejecutar los comandos de nuestro directorio actual mediante **./comando**.

Como separador de los componentes del nombre de un directorio, se usa el carácter **/**. Si usamos un nombre de fichero que empieza por **/**, estamos usando el camino o path completo a ese archivo; esto se llama path absoluto. En cambio, si empieza por **..**, por **...** o por un nombre, estamos usando el path relativo, ya que depende del subdirectorio actual.

Como los directorios también son archivos, si éste no tiene permiso de lectura, no se podrán usar comandos como **ls**; si no tiene permiso de escritura, no se podrán grabar modificaciones a sus archivos ni crear o renombrar archivos, y si no tiene permiso de ejecución en un directorio, no se podrá ingresar a él.

Los nombres de archivos pueden contener caracteres que correspondan a letras, números, el carácter subrayado y el carácter punto. Los ficheros cuyos nombres comienzan por punto permanecen ocultos.

Veamos ahora el nombre de cada directorio de Linux y una breve descripción de su contenido y utilidad (**Figura 2**).

| Directorio | Descripción |
|--------------|---|
| / | Directorio raíz del sistema. |
| /bin | Es la abreviación de "binaries", o ejecutables . Es donde residen la mayoría de los programas esenciales del sistema. |
| /dev | Los "archivos" de /dev son controladores de dispositivo (device drivers) y son usados para acceder a los dispositivos del sistema. |
| /etc | Contiene los archivos de configuración del sistema. |
| /sbin | Se usa para almacenar programas que usará el administrador del sistema. |

| Directorio | Descripción |
|--------------|--|
| /home | Contiene los directorios de los usuarios. Por ejemplo, /home/juan es el directorio del usuario "juan". |
| /lib | Contiene las librerías compartidas. Estos ficheros contienen código que compartirán muchos programas. |
| /proc | Los archivos que contiene residen en memoria, hacen referencia a procesos e información del sistema. |
| /usr | Contienen una serie de subdirectorios con programas y archivos de configuración de usuarios del sistema. |
| /usr/X11R6 | Contiene todos los ejecutables de X Window, ficheros de configuración y de soporte. |
| /usr/local | Es el directorio donde normalmente pondremos nuestras aplicaciones. |
| /usr/src | Contiene el código fuente. El más importante es /usr/src/linux, que contiene el fuente del kernel. |
| /usr/include | Contiene los ficheros de cabecera para el compilador de C. |
| /usr/bin | Contiene la mayoría de los ejecutables que no se encuentran en otras partes, como /bin. |
| /var | Contiene archivos de administración, específicamente históricos del sistema. |

Figura 2. Los directorios estándar de Linux.

Permisos

Por ser un sistema multiusuario, **LINUX** necesita llevar un estricto control de archivos. Para proteger los archivos de cada usuario y aquellos importantes propios del sistema, tales como el archivo de claves, de la manipulación por parte de otros, **LINUX** proporciona un mecanismo conocido como permisos de ficheros, que permite que ficheros y directorios "pertenezcan" a un usuario en particular.



ARCHIVOS EJECUTABLES

Para que un archivo sea ejecutable, sólo debe activarse el atributo de ejecución adecuado.

Por ejemplo, jperez creó ficheros en su directorio `home`, por lo que jperez es su propietario y tiene acceso a ellos. Como vimos en el [Capítulo 4, Usuarios](#), cada uno pertenece por lo menos a un grupo del sistema. Así, jperez puede decidir compartir o no sus archivos con los usuarios pertenecientes a su mismo grupo o con todos los del sistema. Si lo desea, podría restringir el acceso a sus ficheros de forma que ningún otro usuario pueda acceder, excepto el root, por supuesto. De cualquier modo, en la mayoría de los sistemas por defecto se permite que otros usuarios puedan leer nuestros ficheros, pero no modificarlos o borrarlos.

Por lo general, los grupos son definidos por el tipo de usuarios que acceden a la máquina. Por ejemplo, en un sistema de una universidad, los usuarios pueden dividirse en los grupos: estudiantes, dirección, profesores e invitados.

También hay unos pocos grupos definidos por el sistema (como `bin` y `admin`), que son usados para controlar el acceso a los recursos. Muy raramente, los usuarios normales pertenecen a estos grupos.

Los permisos están divididos en tres tipos: lectura, escritura y ejecución, y se identifican con las letras `r`, `w` y `x`, respectivamente. Estos permisos pueden fijarse para tres clases de usuarios: el propietario del fichero, el grupo al que pertenece el fichero y para todos los usuarios independientemente del grupo.

- r Permisón de lectura: permite a un usuario leer el contenido del fichero o, en el caso de un directorio, listar su contenido (usando `ls`).
- w Permisón de escritura: permite a un usuario escribir y modificar el fichero. Para directorios, el permiso de escritura permite crear nuevos ficheros o borrar ficheros ya existentes en él.
- x Permisón de ejecución: permite a un usuario ejecutar el fichero si es un programa o script del shell. Para directorios, el permiso de ejecución permite al usuario cambiar al directorio en cuestión con `cd`.

Interpretación de permisos

Veamos un ejemplo del uso de permisos de ficheros. Usando el comando `ls` con la opción `-l`, se mostrará un listado "largo" de los ficheros que incluye los permisos.

```
[jperez@amankay /jperez]#ls -l  
-rwxrwxrwx 1 jperez users 505 Mr 13 19:05 archivo
```

Analicemos esta línea de directorio. Primero, podemos decir que este fichero pertenece a `jperez` y al grupo `users`. La cadena `-rwxrwxrwx` nos muestra los permisos asignados al archivo: los tres primeros corresponden al propietario; los que le siguen, al grupo; y los últimos, a todos. El primer carácter de la cadena de permisos (`-`) representa el tipo. El `-` significa que es un archivo.

Las siguientes tres letras (`rwx`) representan los permisos para el propietario del fichero, `jperez`, `r` para lectura, `w` para escritura y `x` para ejecución. Los tres caracteres siguientes representan los permisos del grupo; aquí vemos que este archivo tiene permisos de lectura y ejecución del grupo. Los últimos tres caracteres representan los permisos para todos los usuarios, también de lectura, escritura y ejecución.

Dependencias de permisos

Es importante tener en cuenta que los permisos de un fichero también dependen de los permisos del directorio en el que residen. Por ejemplo, aunque un fichero tenga los permisos `-rwxrwxrwx`, otros usuarios no podrán acceder a él a menos que también tenga permiso de lectura y ejecución para el directorio en el cual se encuentra. Si `jperez` quiere restringir el acceso a todos sus ficheros, simplemente podría poner los permisos de su directorio `home/home/jperez` a `-rwx-----`. De esta forma, ningún usuario podrá acceder a su directorio ni a ninguno de sus ficheros o subdirectorios. `Jperez` no necesita preocuparse de los permisos individuales de cada uno de sus ficheros.

En otras palabras, para acceder a un fichero, debemos tener permiso de ejecución de todos los directorios a lo largo del camino de acceso al fichero, además de permiso de lectura (o ejecución) del fichero particular.

Por lo general, los usuarios son muy abiertos con sus ficheros. Los permisos que se dan a los ficheros usualmente son `-rw-r--r--`, lo que permite a todos los demás usuarios leer los ficheros, pero no modifi-

carlos. Los directorios suelen tener los permisos **-rwxr-xr-x**, de modo que los demás usuarios pueden moverse y ver los directorios, pero no borrar o crear nuevos ficheros.

Muchos usuarios tal vez quieran limitar el acceso de otros a sus ficheros. Poniendo los permisos de un fichero a **-rw-----**, no se permite a ningún otro usuario acceder al fichero. Igualmente, poniendo los permisos del directorio a **-rwx-----**, los demás usuarios no podrán acceder al directorio en cuestión.

Cambiando permisos

El comando **chmod** se usa para establecer los permisos de un fichero. Sólo el propietario y el superusuario (root) pueden cambiar los permisos del fichero. La sintaxis de **chmod** es:

```
chmod {a,u,g,o}{+,-}{r,w,x} <archivo>
```

Como ya dijimos, los permisos se separan en tres grupos y se usan letras para referirse a ellos con el comando **chmod**:

- **u**: corresponde al propietario, por lo que se usa la letra u de "user".
- **g**: se refiere al grupo (**g** de group).
- **o o a**: para el resto de los usuarios, podemos usar indistintamente la letra **a** (de all) o la letra **o** (de other).

Para especificar si se están añadiendo o quitando permisos, usamos:

- **+**: para autorizar
- **-**: para desautorizar
- **=**: para resetear

Finalmente, se especifica el tipo de permiso:

- **r**: lectura (de read)
- **w**: escritura (de write)
- **x**: ejecución (de execute)

Algunos ejemplos:

chmod a+r archivo

Da a todos los usuarios acceso al fichero.

chmod +r archivo

Lo mismo que el anterior. Si no se indica **a, u, g u o**, por defecto se toma **a**.

chmod og-x archivo

Quita permisos de ejecución a todos los usuarios, excepto al propietario.

chmod u+rwx archivo

Permite al propietario leer, escribir y ejecutar el fichero.

chmod o-rwx archivo

Quita permisos de lectura, escritura y ejecución a todos los usuarios, menos al propietario y a los usuarios del grupo del fichero.

Existe otro método para asignar o quitar permisos, al que pueden recurrir los usuarios más experimentados. Se trata de la nomenclatura octal, que se basa en usar tres octetos, uno para cada grupo de permisos. Si observamos la **Figura 3**, podremos entender mejor el sistema:

| Binario | Octal | Permisos | Descripción |
|---------|-------|----------|--------------------------------|
| 000 | 0 | --- | ningún permiso |
| 001 | 1 | --x | ejecución |
| 010 | 2 | -w- | escritura |
| 011 | 3 | -wx | escritura y ejecución |
| 100 | 4 | r-- | lectura |
| 101 | 5 | r-x | lectura y ejecución |
| 110 | 6 | rw- | lectura y escritura |
| 111 | 7 | rwx | lectura, escritura y ejecución |

Figura 3. Permisos de archivos en formato binario y octal.

Como se puede apreciar, la numeración octal se adecua perfectamente. Así, podemos usar tres números octales para asignar los permisos. Veamos algunos ejemplos.

chmod 700 programa

7=111=rwx: corresponde al propietario y tiene todos los permisos.

0=000=---: corresponde al grupo y no tiene ningún permiso.

0=000=---: corresponde al resto y tampoco tiene permisos.

chmod 644 carta

6=110=rw-: corresponde al propietario y tiene permisos de lectura y escritura.

4=100=r- -: corresponde al grupo y únicamente tiene permiso de lectura.

4=100=r- -: corresponde al resto y tiene sólo permiso de lectura.

chmod 755 programa

7=111=rwx: corresponde al propietario y tiene todos los permisos.

5=101=r-x: corresponde al grupo y tiene permisos de lectura y ejecución.

5=101=r-x: corresponde al resto y tiene permisos de lectura y ejecución.

chmod 750 programa

7=111=rwx: corresponde al propietario y tiene todos los permisos.

5=101=r-x: corresponde al grupo y tiene permisos de lectura y ejecución.

0=000=- -: corresponde al resto y no tiene ningún permiso.

PROCESOS

| | |
|--|------------------|
| Definición | .Pág. 99 |
| Tabla de procesos en UNIX | .Pág. 99 |
| Creación de procesos | .Pág. 101 |
| Control de tareas | .Pág. 103 |
| Detención y relanzamiento | .Pág. 106 |

“... en Babilonia me quisiste perder en un laberinto de bronce con muchas escaleras, puertas y muros; ahora el poderoso ha tenido a bien que te muestre el mío, donde no hay escaleras que subir ni puertas que forzar; ni fatigosas galerías que recorrer ni muros que te veden el paso.

Luego le desató las ligaduras y lo abandonó en mitad del desierto donde murrió de hambre y de sed.”

Los dos reyes y los dos laberintos, Jorge Luis Borges

Definición

Si por programas entendemos archivos con secuencias de código que indican al microprocesador de la computadora las instrucciones que deberá ejecutar para realizar determinadas tareas, entonces podemos decir que los programas son procesos en potencia.



HAY QUE SABERLO

ORDEN PS AUX

Con la orden ps aux podemos ver todos los procesos que está ejecutando nuestra PC.

Se puede definir a los procesos como programas en tiempo de ejecución, es decir que un proceso es un programa vivo. Hay que destacar que un programa puede convertirse en más de un proceso, y a cada uno se lo denomina "instancia de un programa".

Simplificando, cuando invocamos un programa, al ejecutarse se convierte en un proceso. Cada proceso nuevo es una bifurcación de un proceso padre. Estas bifurcaciones se obtienen a través de la invocación de un servicio del kernel, **fork**.

La ejecución del kernel durante el arranque de LINUX posee capacidades de administración y control de procesos. Muchos usuarios novatos no aprovechan esta facultad, que se relaciona estrechamente con la característica multitarea del LINUX.

Tabla de procesos en UNIX

LINUX, al igual que UNIX, utiliza una tabla llamada tabla de procesos, en la que se centraliza la información de cada uno. Esta tabla se

encuentra compuesta por filas, cada una de las cuales contiene información sobre un solo proceso y es de dimensión fija, por lo que la cantidad de procesos que pueden ejecutarse concurrentemente es limitada. El siguiente es un ejemplo de esta tabla, que se puede consultar mediante el comando **ps**.

| USER | PID | %CPU | %MEM | SIZE | RSS | TTY | STAT | START | TIME | COMMAND |
|--------|-----|------|------|------|-----|-----|------|-------|------|-----------------------|
| daemon | 98 | 0.0 | 0.4 | 840 | 296 | ? | S | 15:21 | 0:00 | /usr/sbin/atd |
| rmt | 1 | 0.0 | 0.4 | 756 | 272 | ? | S | 15:21 | 0:03 | init [2] |
| rmt | 2 | 0.0 | 0.0 | 0 | 0 | ? | SW | 15:21 | 0:00 | (kflushd) |
| root | 3 | 0.0 | 0.0 | 0 | 0 | ? | SW< | 15:21 | 0:00 | (kswapd) |
| rmt | 12 | 0.0 | 0.1 | 720 | 104 | ? | S | 15:21 | 0:00 | update |
| root | 79 | 0.0 | 0.6 | 892 | 404 | ? | S | 15:21 | 0:00 | /sbin/syslogd |
| rmt | 81 | 0.0 | 0.3 | 752 | 248 | ? | S | 15:21 | 0:00 | /sbin/klogd |
| rmt | 90 | 0.0 | 0.3 | 748 | 220 | ? | S | 15:21 | 0:00 | /usr/sbin/gpm -m /dev |
| rmt | 101 | 0.0 | 0.5 | 852 | 372 | ? | S | 15:21 | 0:00 | /usr/sbin/cron |
| rmt | 106 | 0.0 | 0.4 | 836 | 288 | 3 | S | 15:21 | 0:00 | /sbin/getty 38400 tty |
| rmt | 107 | 0.0 | 0.4 | 836 | 300 | 4 | S | 15:21 | 0:00 | lsbingetty 38400 tty |
| root | 108 | 0.0 | 0.4 | 836 | 296 | 5 | S | 15:21 | 0:00 | /sbin/getty 38400 tty |
| rmt | 109 | 0.0 | 0.4 | 836 | 296 | 6 | S | 15:21 | 0:00 | /sbin/getty 38400 tty |

Qué información guarda cada una de las líneas

- PID: (Process ID) es el número único identificatorio del proceso. Cada proceso que se ejecuta tiene asignado un número, que se usa para todas las operaciones relacionadas con él.
- PPID: es el PID del proceso pariente o padre, es decir quien causó la bifurcación para crear este nuevo proceso. Se usa para avisar al parente acerca de la terminación del "hijo".
- Mem: contiene información acerca de los recursos de memoria del proceso, punto de carga, requerimiento total, etc.
- Status: aquí podemos definir tres categorías posibles:
 1. Listo para correr, esperando su turno.
 2. Corriendo en este momento. Sólo hay un proceso en este estado.
 3. Esperando que complete lectura o escritura de disco. En este caso, el proceso se define como bloqueado y no puede correr hasta que se desbloquee.
- Tiempos: tiempos de CPU, tiempo de bloqueo y tiempo total consumidos por el proceso.
- Archivos: información acerca de los archivos abiertos que pueda tener el proceso.

Cada vez que se crea un proceso, LINUX localiza una fila desocupada de la tabla de procesos y coloca allí la información concerniente al proceso recién creado. La fila se va modificando a medida que la ejecución del proceso continúa. Al morir el proceso, la fila se desocupa y queda lista para recibir otro.

Creación de procesos

Para crear un proceso en LINUX, los programadores usan una instrucción que invoca una función del kernel, `fork()`. Veamos a continuación una secuencia completa de creación de un proceso:

1. El proceso creador (padre) ejecuta la llamada a la función `fork()`.
2. El kernel del sistema operativo crea un clon del parente, el hijo, y lo hace correr independientemente.

3. El hijo, que es una copia exacta del padre, procede a "abdicar" a favor del programa que se quería ejecutar como proceso, por medio de otra llamada al sistema `exec()`.

Luego de `fork()`, se utiliza la llamada `exec()` para que se comience a ejecutar un nuevo programa que reemplaza al hijo. Veamos un gráfico que nos servirá para aclarar esta idea (**Figura 1**):

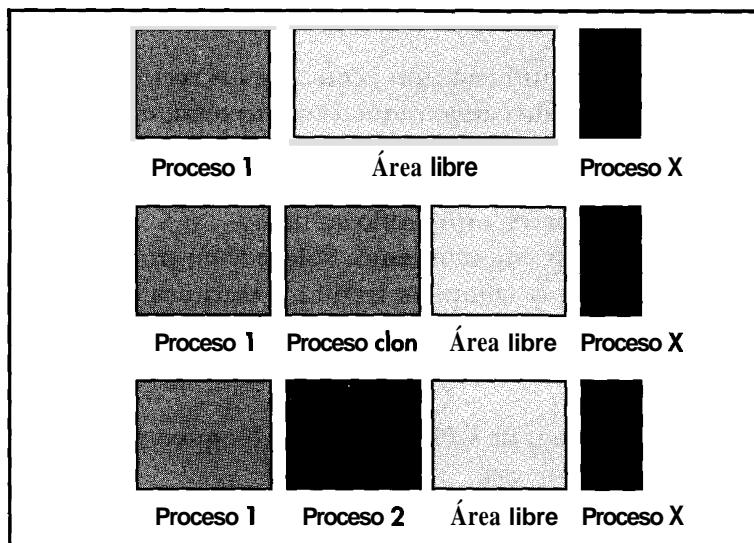


Figura 1. Secuencia de creación de un nuevo proceso.

Vemos que, en un primer tiempo, tenemos un proceso 1 en color rojo, el área libre en color amarillo y el resto de los procesos, identificados como proceso x, en color azul.

En el segundo tiempo, cuando ya se ejecutó la llamada a la función `fork()`, tenemos el proceso 1 y su clon, el proceso 1 clon, ambos en rojo; obviamente el área libre en color amarillo y el resto de los procesos en azul como proceso x.

Finalmente, en un tercer tiempo, tenemos el proceso 1 en color rojo, el proceso 2 en color verde, que ya no es un clon de su padre, el proceso 1, sino que hemos efectuado una llamada a la función `exec()`, mediante la cual creamos un nuevo proceso. También tenemos el resto del área libre en color amarillo y los procesos restantes en el clásico color azul identificados como proceso x.

Para quienes tengan mínimos conocimientos de programación en C, veamos un pequeño ejemplo que ilustrará mejor este proceso. Antes de proceder, debemos aclarar que la llamada a la función **fork()** devuelve cero si es ejecutada por el padre, y un valor no cero si es ejecutada por el hijo.

```
/* ---- Ejemplo de creación de proceso ---- */
n = fork();      /*padre ejecuta el fork()*/
if ( n != 0 ) { /* proceso padre! */
    wait(&status) /* espera a su hijo */
} else{          /* el hijo se crea */
    exec("programa"...); /* ejecución de nuevo prog. */
}
```

Control de tareas

La capacidad multitarea de LINUX permite a los usuarios correr más de un programa a la vez. Supongamos que tenemos que ejecutar un programa que realiza tareas sin necesidad de interactuar con el usuario. En este tiempo que es tiempo muerto de terminal, el usuario podría continuar trabajando en otras tareas. Aparentemente, la única solución es esperar, pero en el mundo de LINUX, esto no es así, ya que podemos pasar esa tarea a segundo plano (background) y retomar el control del teclado y pantalla para lanzar un nuevo proceso, que se ejecutará en primer plano (foreground).



MÁS DATOS

COMANDOS BG Y FG

Los comandos que nos permiten la ejecución de tareas en segundo y primer plano son **bg** y **fg**.

Para exemplificar esto, veamos un caso muy común que se usa para demostrar el control de tareas.

El comando **yes** es un comando aparentemente inútil que envía una serie interminable de **y-es** a la salida estándar. En realidad, es muy útil. Si se utiliza una tubería (o "pipe") para unir la salida de **yes** con otro comando que haga preguntas del tipo si/no, la serie de **y-es** confirmaría todas las preguntas.

```
/home/jperez# yes  
Y  
Y  
Y  
Y  
y CONTROL+C
```

La serie de **y-es** continuará hasta el infinito, a no ser que la detengamos con CONTROL+C. También podemos deshacernos de esta serie de **y-es** redirigiendo la salida estándar de **yes** hacia **/dev/null**, que es como una especie de "aguero negro" o dispositivo nulo. Todo lo que usted envíe allí, desaparece.

```
/home/jperez# yes > /dev/null
```

Ahora no vemos **y-es** en pantalla porque están redirigidos al dispositivo nulo. Pero el prompt del shell no retorna. Esto es porque **yes** se sigue ejecutando en primer plano. Para recuperarlo, pulsamos la tecla de interrupción CONTROL+C.

Supongamos ahora que queremos dejar que el comando **yes** se siga ejecutando, y volver al mismo tiempo al shell para trabajar en otras tareas. Para ello enviaremos **yes** a segundo plano, lo que nos permitirá ejecutarlo, pero sin necesidad de interacción con el usuario.

Una forma de mandar procesos a segundo plano es añadiendo un carácter **&** al final de cada comando.

```
/home/jperez# yes > /dev/null &
```

```
[1] 164
```

Como podemos ver, ha regresado al prompt del shell. [1] representa el número de tarea del proceso **yes**. El shell asigna un número a cada

tarea que se esté ejecutando. Como **yes** es el único comando en esta situación, se le asigna el número de tarea **1**. El numero **164** identifica el proceso, o PID, el número que el sistema le asigna al proceso. Ambos números pueden usarse para referirse a la tarea.

Ahora ya tenemos el proceso **yes** corriendo en segundo plano, y enviando constantemente la señal hacia el dispositivo **/dev/null**. Para chequear el estado del proceso, utilizamos el comando interno del shell **jobs**:

```
/home/jperez# jobs  
[1]+ Running yes >/dev/null &
```

También podemos usar el comando **ps**, como mostramos antes, para comprobar el estado de la tarea.

Para eliminar una tarea, se utiliza el comando **kill**. Este comando toma como argumento un número de tarea o un número de PID de un proceso.

```
/home/jperez# kill %1
```

Cuando se identifica la tarea con el número de tarea, se debe pre-ceder el número con el carácter de porcentaje (%).

Usemos nuevamente el comando **jobs** para comprobarlo:

```
/home/jperez# jobs  
[1]+ Terminated yes >/dev/null
```

La tarea está muerta, y si usamos el comando **jobs** de nuevo, ya no mostrará nada.

También podemos matar la tarea usando el número de PID:

```
/home/jperez# kill 164  
es equivalente a
```

```
/home/jperez# kill %1
```

No es necesario usar el % cuando nos referimos a una tarea a través de su PID.

Detención y relanzamiento

Supongamos que lanzamos el programa BitchX, que es un cliente de IRC (chat), y mientras conversamos, alguien nos pide un archivo que nosotros sabemos que tenemos, pero no recordamos dónde. En este caso, podemos detener el proceso correspondiente a BitchX pulsando CONTROL+Z y así retomar el control de shell para buscar el archivo en cuestión.

Una vez terminada la búsqueda, retomamos el control del proceso detenido poniéndolo nuevamente en primer plano mediante el comando **fg**. Pero el programa relegado a segundo plano no está en ejecución, sino que, por decirlo de una manera gráfica, fue puesto en pausa. Si queremos que continúe con su tarea, debemos usar el comando **bg**, que le da la orden de retomar su ejecución en segundo plano.

Sólo puede haber un proceso en primer plano, el que interactúa con el usuario mediante la entrada estándar (teclado) y la salida estándar (pantalla), salvo obviamente que se haya redirigido la salida.



MÁS DATOS

TÉRMINOS PROCESO Y TAREA

Un proceso que está corriendo se denomina tarea para el shell. Los términos proceso y tarea son sinónimos. Sin embargo, se suele denominar "tarea" a un proceso cuando nos referimos al control de tareas.

Hay que tener en cuenta que suspender un trabajo no es lo mismo que interrumpirlo. Cuando interrumpimos un proceso (generalmente con la pulsación de CONTROL+C), el proceso muere, y deja de estar en memoria y de utilizar recursos del equipo. Una vez eliminado, el proceso no puede continuar ejecutándose. También se puede dar el caso de que algunos programas capturan la interrupción, de modo que pulsando CONTROL+C, no se detenga. Esto se hace para que el programa pueda realizar operaciones necesarias de limpieza antes de terminar.

Volvamos a nuestro ejemplo con **yes**. Lanzamos el proceso **yes** en primer plano:

```
/home/jperez# yes > /dev/null
```

Dado que **yes** corre en primer plano, no retorna el prompt del shell. Ahora, suspendemos la tarea con CONTROL+Z. El suspender

una tarea no la mata: solamente la detiene en forma temporal hasta que decidamos retomarla.

```
/home/jperez# yes > /dev/null
```

...

```
CONTROL+Z
```

```
[1]+ Stopped yes >/dev/null
```

```
/home/jperez#
```



MÁS DATOS

COMANDO FG

Sé puede usar fg tanto con tareas detenidas como con las que están en segundo plano.

Hay una gran diferencia entre una tarea que se encuentra en segundo plano y una que está detenida. Una tarea detenida es una tarea que no se está ejecutando, es decir, que no usa tiempo de CPU y que no está haciendo ningún trabajo.

En cambio, una tarea en segundo plano sí se está ejecutando, y está usando memoria y CPU.

Sin embargo, una tarea en segundo plano puede intentar mostrar texto en su terminal, lo que puede resultar molesto si estamos haciendo otra cosa. Por ejemplo, si usamos el comando

```
/home/jperez# yes &
```

sin redirigir la salida a `/dev/null`, una cadena de y-es se mostrará en el monitor, sin que podamos interrumpirlo (no podemos hacer uso de CONTROL+C para interrumpir tareas en segundo plano). Para poder parar esas interminables y-es, tendríamos que usar el comando `fg` para pasar la tarea a primer plano y, recién entonces, usar CONTROL+C para matarla.



HAY QUE SABERLO

COMANDOS FG Y BG

Normalmente, los comandos `fg` y `bg` actúan sobre el último proceso parado (indicado por un + junto al número de tarea cuando usamos el comando `jobs`).

Si tenemos varios procesos corriendo a la vez, podremos mandar a primer o segundo plano una tarea específica indicando el ID.



MÁS DATOS

EQUIVALENCIAS

Si usamos el número de tarea por sí sólo, como /home/jperez# %2 es equivalente a /home/jperez# fg %2) de tarea como argumento de fg o bg.



HAY QUE SABERLO

EXCEPCIONES

No se pueden usar los PID de proceso con fg o bg.

Sólo queda recordar que el uso de control de tareas es una utilidad del shell. Los comandos fg, bg y jobs son internos del shell.

APLICACIONES

| | |
|--------------------------|------------------|
| Paquetes | Pág. 111 |
| Paquetes TGZ | .Pág. 111 |
| Paquetes DEB | .Pág. 111 |
| Paquetes RPM | .Pág. 112 |
| Usando RPM | .Pág. 113 |
| Compilación | .Pág. 114 |

"Todo lo que una persona puede imaginar; otras podrán hacerlo realidad."

Julio Verne

Paquetes

Con la intención de facilitar la distribución, instalación y mantenimiento del software, varias instituciones han creado sistemas de empaquetamiento. Podemos decir que un paquete de software es un objeto que contiene en sí mismo el software a instalar, como así también toda la información necesaria para hacerlo, esto es: la ubicación que tendrán los archivos en el disco rígido, la información acerca de la versión del mismo, las interdependencias con otros paquetes, la cantidad de espacio de disco requerida por el software, etc.

Cuando hablamos de dependencias, nos referimos a la dependencia que tiene un paquete de otro que esté previamente instalado. Por ejemplo, si queremos instalar un juego que va a hacer uso de las librerías svga, se dice que el paquete del juego depende del paquete de la librería, por lo que ésta debe encontrarse instalada previamente; en caso contrario, el manejador de paquetes no nos permitirá la instalación. Como veremos más adelante, este requisito puede ser obviado.

Paquetes TGZ

Lamentablemente, no existe un estándar para todas las distribuciones de LINUX, es decir que nos encontramos con que Slackware usa el formato **tgz**, que no es otra cosa que archivos y directorios empaquetados con tar y comprimidos con gzip. En este caso, el paquete únicamente almacena los archivos, y para que la distribución por los directorios sea correcta, se debe descomprimir el paquete desde la raíz.

Veamos un ejemplo:

```
% tar xvfv XF86_S3V.tgz
```

Paquetes DEB

La distribución Debian tiene un formato propio de paquetes; es muy similar en esencia al sistema de Red Hat, pero se necesitan otras

herramientas tanto para su creación como para su manipulación. También mantiene información extra sobre el software a instalar y un completo registro de dependencias.

Veamos un ejemplo:

```
dpkg -i xf86-SVGA .deb
```

Paquetes RPM

Red Hat es el creador del formato RPM (Red Hat Packet Manager), uno de los más completos y exitosos, además de ser libre y abierto para su uso en otras distribuciones, por lo que fue adoptado por S.U.S.E y Eurelek, entre otras. Lleva registros de dependencias, versiones, etc. y también mantiene una completa base de datos acerca del software instalado, por lo que su seguimiento y ubicación resultan muy sencillos.



MÁS DATOS

UN LIBRO SOBRE ESTE SISTEMA

En www.redhat.com podemos encontrar un libro de Ed Bailey llamado Maximum RPM en el que encontraremos más información acerca de este sistema.



HAY QUE SABERLO

VISUALIZAR TODOS LOS PARÁMETROS POSIBLES

Para ver todos los parámetros posibles, podemos usar **--help** o consultar la página del manual con **man rpm**.

Veamos a continuación un detalle reducido de sus opciones :

| Opción | Descripción |
|------------------|---|
| --help | Muestra la ayuda |
| --version | Muestra la versión del manejador de paquetes |
| --root | Permite establecer un directorio raíz diferente a / |
| -q | Permite hacer consultas sobre el paquete |
| -i | Instala el paquete |
| -e | Borra el paquete instalado |

| Opción | Descripción |
|---------------|---|
| -U | Actualiza el paquete |
| --force | Fuerza algún proceso |
| --nodeps | No verifica dependencias |
| -R | Lista las dependencias del paquete |
| --excludedocs | No instala la documentación del paquete |
| -b | Crea paquetes RPM |

La utilidad **rpm** cuenta, por supuesto, con su página de manual.

Usando RPM

Instalación de paquetes RPM

rpm -i comunicator-4.07-i386.rpm

Instala el paquetes **comunicator-4.07-i386.rpm**, que contiene el ejecutable de Netscape Communicator versión 4.07 para LINUX plataforma i386, es decir PCs. Además actualiza automáticamente la base de datos que lleva el registro de paquetes instalados.

Borrado o desinstalación de paquetes RPM

rpm -e ccmnicator-4.07

Una vez ejecutada esta linea, los archivos correspondientes a Netscape Communicator versión 4.07 para LINUX son borrados de nuestro disco rígido y se actualiza la base de datos de paquetes instalados.

Actualización de paquetes RPM

rpm -U comunicator-4.07-i386.rpm

Con este linea, lo que en realidad hacemos es borrar cualquier versión anterior del paquete e instalar la nueva, previa verificación de dependencias.

Compilación

Una de las características sobresalientes de las aplicaciones para LINUX es que se distribuyen con sus fuentes de acuerdo con lo establecido por la licencia GNU. La mayoría de las veces, nos encontraremos con paquetes precompilados listos para su uso, pero puede suceder que no estén disponibles, por lo que deberemos compilar el programa para poder aprovecharlo.

Supongamos que queremos instalar los fuentes del programa BitchX, que es un cliente de IRC, y queremos compilarlo e instalarlo.



MÁS DATOS

WWW

La página oficial de BitchX es www.bitchx.org.

Primero, necesitaremos obtener estos fuentes, para lo cual debaremos bajar de Internet el archivo ircii-pana-75p1.tar.gz. Luego, creamos en nuestro disco rígido un subdirectorio, en el que colocaremos los fuentes para su compilación. Una buena práctica es crear un directorio llamado, por ejemplo, **taller**, **laboratorio** o **instalaciones**, de donde colgarán todos los subdirectorios con fuentes de programas a compilar.



MÁS DATOS

Por defecto, la instalación del código fuente se hace en el directorio /usr/src.

Ahora ingresamos al directorio **taller** y descomprimimos el archivo que obtuvimos de Internet.

```
% tar xvfz ircii-pana-75p1.tar.gz
BitchX/
BitchX/configure
BitchX/include/
BitchX/include/config.h
BitchX/include/array.h
BitchX/include/alias.h
BitchX/include/chelp.h
BitchX/include/whowas.h
```

```
BitchX/include/misc.h  
BitchX/include/crypt.h  
BitchX/include/ctcp.h  
BitchX/include/dcc.h  
BitchX/include/debug.h  
BitchX/include/defs.h.in  
...continúa...
```

Bien, ya tenemos en nuestro disco los archivos para compilar, pero antes veamos qué nos dice el autor de este programa acerca de él y del proceso de instalación. Casi siempre encontraremos entre los archivos uno llamado README y otro INSTALL, aunque a veces están los dos incluidos en README. Al leerlos, encontraremos toda la información necesaria para la compilación y posterior instalación del programa. En este caso en particular, tenemos INSTALLATION; lo miramos con:

```
% less INSTALLATION
```

Vemos que contiene información sobre las plataformas soportadas, una breve introducción y los pasos necesarios para la compilación.

Estando ubicados en el directorio **/taller/BitchX/**, procedamos a ejecutar el script de configuración; este paso es necesario para reconocer la plataforma en la que se compilará. El script verificará la existencia del compilador y las librerías necesarias y, finalmente, creará los archivos que usará la herramienta **make**.

```
% ./configure  
creating cache ./config.cache  
Welcome to BitchX-75p1 configuration.  
After this configure is done, type make in order to compile BitchX-75p1  
checking for gcc... gcc  
checking whether the C compiler (gcc ) works... yes  
checking whether the C compiler (gcc ) is a cross-compiler... no  
checking whether we are using GNU C... yes  
checking whether gcc accepts -g... yes  
checking how to run the C preprocessor... gcc -E  
checking for POSIXized ISC... no  
checking for AIX...  
...continúa...
```

Puede suceder que se nos informe de la falta de alguna librería, en cuyo caso tendremos que recurrir nuevamente a Internet para obtenerla.

Si todo salió bien (es lo más probable), podemos comenzar el proceso de compilación, que se hace con:

```
% make  
make[1]: Entering directory `/BitchX/source'  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c alias.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c alist.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c array.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c banlist.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c bot_link.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c compat.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c cdcc.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c chelp.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c commands.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c commands2.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c crypt.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c ctcp.c  
gcc -O2 -Wall -D_VERSION_=\"BitchX\" -I../include -c cset.c  
...continua...
```

Este paso es el que más tiempo nos llevará. Una vez terminado, dentro del directorio **/taller/BitchX/src** tendremos el ejecutable de BitchX. Sólo nos resta instalarlo y disfrutar del mejor cliente de chat. Para la instalación, ingresaremos desde la línea de comandos:

```
% make install
```

Listo. Con esto ya hemos descomprimido, compilado e instalado el programa. Este procedimiento puede variar dependiendo de cada programador, pero siempre sigue una secuencia similar. No olvidemos leer la documentación que acompaña al programa.

KERNEL

| | |
|---|-----------------|
| ¿Qué es el kernel? | Pág. 119 |
| ¿Por qué recompilar el kernel? | Pág. 119 |
| Los fuentes del kernel | Pág. 119 |
| Estables y experimentales | Pág. 120 |
| Preparación | Pág. 121 |
| Configuración | Pág. 121 |
| Code maturity level options | Pág. 122 |
| Loadable module support | Pág. 123 |
| General setup | Pág. 123 |
| Floppy, IDE and other blocks devices | Pág. 125 |
| Networking options | Pág. 127 |
| SCSI support | Pág. 127 |
| SCSI low-level drivers | Pág. 128 |
| Network device support | Pág. 128 |
| ISDN support | Pág. 130 |
| CD-ROM drivers | Pág. 130 |
| Filesystems | Pág. 130 |
| Character devices | Pág. 132 |
| Sound | Pág. 132 |
| Kernel hacking | Pág. 133 |
| Compilación | Pág. 133 |
| LILO | Pág. 134 |

"-Maestro, quisiera saber cómo viven los peces en el mar.

-Como los hombres en la tierra: los grandes se comen a los pequeños."

William Shakespeare

¿Qué es el kernel?

El kernel es el núcleo de todo sistema operativo, es decir que es el binario que se ejecuta en primera instancia y se aloja en memoria para desarrollar sus tareas durante todo el tiempo que dure la sesión.

¿Por qué recompilar el kernel?

Existen muchas razones para recompilar el kernel. Fundamentalmente, lo haremos para adecuarlo al hardware de nuestra PC, ya que la versión de la que disponemos no puede manejar cierto tipo de hardware que tenemos instalado, sino que hace falta una actualización a una versión superior. Uno de los grandes problemas con que se encuentra el usuario novato es que no cuenta en el kernel con el soporte necesario para determinado hardware. En estos casos, lo mejor es actualizar los fuentes del núcleo. Con la actualización del kernel, también obtenemos mejoras del rendimiento del sistema; ajustando al máximo la configuración, un núcleo configurado con características que no va a usar será más grande en tamaño y, en consecuencia, consumirá más recursos y ralentizará el sistema.

Una ventaja extra que obtenemos es poder contar con varias versiones del kernel para bootear, de acuerdo con lo que queramos hacer con la PC. Por ejemplo, podemos tener dos versiones, una para el trabajo normal y otra para conectarse a Internet, en la que incluiremos soporte de red, PPP, protección contra *buffer overflow*, etc.

Los fuentes del kernel

Dependiendo de la versión del núcleo, tenemos diferentes archivos que lo conforman. Éstos se encuentran distribuidos en un árbol de directorio bastante extenso que, por lo general, cuelga del directorio `/usr/src/linux`. También puede suceder que LINUX sea un enlace

simbólico que apunte al verdadero directorio, el que normalmente se denomina con la siguiente nomenclatura: linux-2.0.36 o linux-2.1.130. A continuación, presentamos una breve descripción de cada directorio, que puede variar entre versiones.



WWW

La página oficial del kernel es www.kernel.org.

[MÁS DATOS](#)

arch: contiene el código de los ficheros que configuran LINUX para ser ejecutado en distintas arquitecturas. En este caso, disponemos de Alpha, Mips, Sparc e i386 para los procesadores de la familia Intel. En el resto de la estructura de ficheros, tenemos los fuentes para el LINUX 'genérico', por decirlo de alguna forma, en el que están los ficheros que marcan la diferencia para las distintas arquitecturas.

drivers: aquí disponemos de la parte del núcleo destinada al manejo de dispositivos.

fs: son los sistemas de ficheros disponibles (minix, msdos, ext2, etc.).

include: ficheros cabecera organizados según la arquitectura a la que están destinados.

init: contiene el fichero principal del código fuente (main.c).

ipc: implementa los distintos mecanismos destinados a la comunicación entre procesos que soporta LINUX.

kernel: fuentes del núcleo común a todas las arquitecturas.

lib: ficheros con los que se construye la librería estática lib.a, que posteriormente se incorpora al núcleo.

mm: gestión de la memoria en LINUX.

net: implementación de los protocolos que el sistema operativo puede manejar en cuanto a redes y comunicaciones se refiere.

E y i |

En el desarrollo del núcleo de LINUX, encontramos que tiene dos flujos de desarrollo. Por una parte, el que usa la fórmula x.par.xxx y, por otro, x.impar.xxx. Esto es una convención que nos permite identificar los kernels estables de los experimentales. Así, los que tengan como segundo número un valor par son estables, es decir que ya han sido lo suficientemente probados y corregidos. Los experimentales son

los que se usan para probar nuevas tecnologías, drivers para nuevo hardware, correcciones, etc. Es mi recomendación usar los estables a menos que sea absolutamente necesario usar los otros, como puede suceder si precisamos que reconozca algunas placas de sonido PnP.



HAY QUE SABERLO

Si bien la distribución R. H. 5.2 trae el Kernel 2.0.36, al momento de terminar este libro, la última versión estable es la 2.2.1.

Preparación

De toda compilación previa queda una serie a archivos y "basura" en general, además de archivos de configuración que se deben eliminar al momento de emprender una nueva compilación. Esto se hace de la siguiente manera:

```
[linux1 #make mrproper  
make[1]: Entering directory `/usr/src/linux-2.0.36/arch/i386/boot'  
rm -f bootsect setup  
rm -f bbootsect  
rm -f zImage tools/build compressed/vmlinux.out  
...continúa ...
```

I Ó

Aunque existe una nueva versión estable del Núcleo, analizaremos la 2.0.36 por ser la que se distribuye con R. H. 5.2.

Existen varios métodos para realizar la configuración. El más antiguo y ya en desuso se invoca mediante **make config**. El segundo, que se utiliza cuando no se cuenta con una terminal gráfica, es **make menuconfig**, que nos presenta un sistema de menús y cuadros de diálogo en modo texto. Finalmente, el sistema de configuración gráfico, **make xconfig**, es el que veremos en detalle. De todas formas, lo aquí expuesto es de aplicación también para los métodos anteriores.

Lo primero que tenemos que hacer es invocar a la compilación del programa de compilación desde una terminal gráfica. Éste se ejecuta automáticamente.

Una vez terminada la compilación, podremos apreciar que en nuestra pantalla aparece un nueva ventana como la que muestra la **Figura 1**.

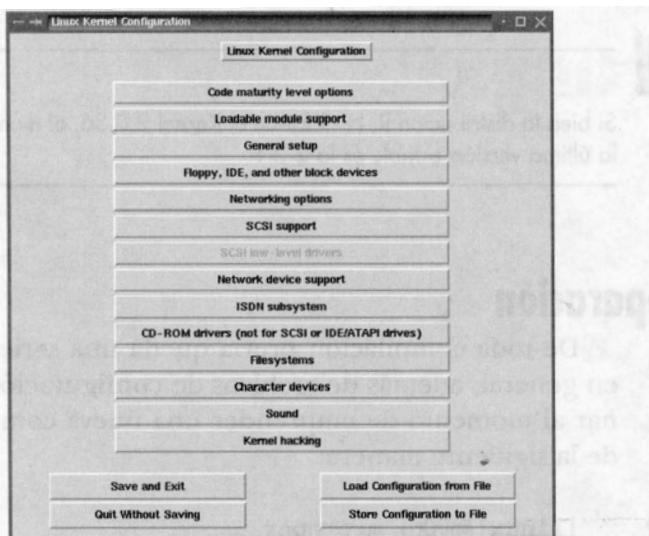


Figura 1. El menú principal con todas la secciones que habrá que configurar para la compilación.

Code maturity level options

Varios de los elementos que soporta LINUX, tales como sistemas de archivos, drivers para determinado hardware, protocolos de comunicaciones, etc., se encuentran en estado de desarrollo. Sin embargo, la configuración nos permite incorporar estas capacidades experimentales en la compilación. A menos que necesitemos algún protocolo o driver especial, es recomendable contestar negativamente en este punto, ya que podría significar un kernel inestable (**Figura 2**).

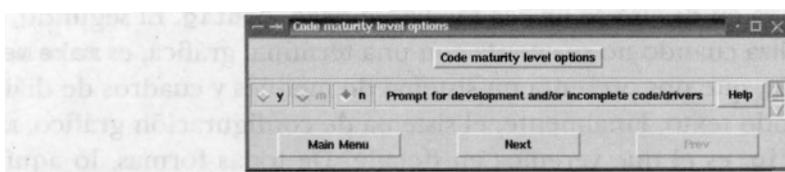


Figura 2. Las cualidades experimentales del kernel pueden tornarlo inestable.

Loadable module support

Los módulos del kernel son pequeñas piezas de código que pueden ser compiladas en forma separada del kernel, para luego, si es necesario, instalarlas o removérlas usando los programas `insmod` y `rmmod`. Supongamos que tenemos que usar ocasionalmente un Zip drive externo. No tiene sentido compilar el kernel con soporte permanente; en este caso, así como en otros similares, es recomendable compilar el soporte para Zip en forma de módulo para poder incorporarlo al kernel sólo cuando sea necesario (**Figura 3**). No es recomendable usar módulos, a menos que sea indispensable.

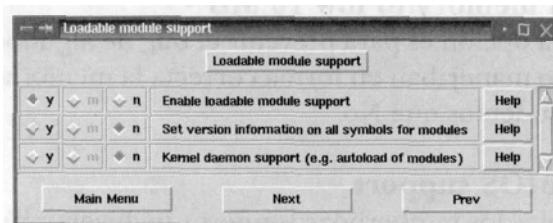


Figura 3. Los módulos son una buena opción en el caso de necesitar algún soporte sólo ocasionalmente.

General setup

En este punto, nos encontramos con opciones generales acerca del kernel que vamos a generar. En la **Figura 4** podemos ver las opciones de la configuración general del kernel.

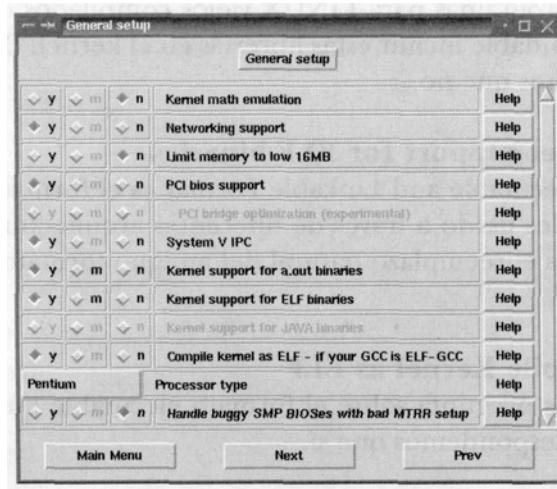


Figura 4. Opciones generales.

Kernel math emulation

En caso de no disponer de un coprocesador matemático, el kernel puede emularlo para las operaciones de punto flotante. A menos que nuestra PC sea una vieja 486, debemos contestar No.

Networking support

Aquí debemos responder afirmativamente aun cuando nuestra PC no se encuentre en red. A menos que estemos seguros de lo que estamos haciendo, no responderemos en forma negativa.

Limit memory to low 16 MB

Esta opción es para prevenir el bug de algunos viejos motherboards que no manejaban en forma correcta la memoria por sobre los 16 MB. Respondemos que No.

PCI BIOS support

Todos los motherboards nuevos incluyen ranuras PCI, que es un tipo de bus del sistema. Contestamos afirmativamente.

System V IPC

Es una librería de funciones y llamadas al sistema para la sincronización e intercomunicación de procesos; contestamos que sí.

Kernel support for a.out binaries

A.out es el antiguo formato de ejecutables de LINUX. Si contamos con programas para LINUX viejos compilados en este formato es recomendable incluir estas librerías en el kernel. De no ser así, sólo contestamos que no.

Kernel support for ELF binaries

Executable and Linkable Format es el formato para librerías y ejecutable usado a través de diferentes arquitecturas y sistemas operativos. Es el reemplazo natural del antiguo formato a.out. Respondemos que sí.

Compile Kernel as ELF

Nos pregunta sobre el formato ejecutable que tendrá nuestro kernel. Respondemos que si.

Processor type

Indicamos el tipo de procesador de nuestra PC. Esto sólo se usa con fines de optimización.

Handle buggy SMP BIOSes whit bad MTRR setup

Esto es para prevenir el bug de algunos Pentium Pro. A menos que lo tengamos, respondemos que no.

Floppy, IDE and other blocks devices

En este punto de la configuración, nos encargaremos de los dispositivos de almacenamiento magnético, de las interfaces y de otras características de las placas madre (**Figura 5**).

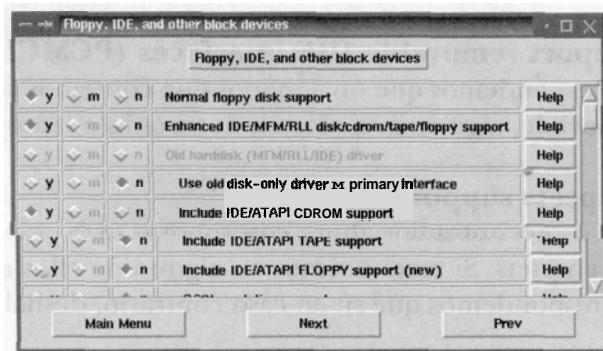


Figura 5. Seleccionamos los dispositivos adecuados.

Normal floppy disk support

Respondemos que sí para poder acceder a disquetes.

Enhanced IDE/MFM/RLL...

Para un soporte completo de dispositivos IDE, como el manejo de discos mayores que 549 MB, es necesario incluir esta opción. Contestamos que sí.

Use old disk-only driver on primary interface

Para usar el driver viejo sólo para discos como interfase IDE primaria. Respondemos que no.

Include IDE/ATAPI CDROM support

Respondemos que sí en caso de tener una lectora de CD-ROM norma ATAPI.

Include IDE/ATAPI TAPE support

Contestamos afirmativamente si tenemos un tape backup norma ATAPI.

Include IDE/ATAPI FLOPPY support

Respondemos que sí en caso de tener un floppy norma ATAPI.

SCSI emulation support

Produce una emulación SCSI para dispositivos IDE/ATAPI. Respondemos que no.

Support removable IDE interfaces (PCMCIA)

Respondemos que no a menos que contemos con algún dispositivo removable, algo común sólo en notebooks o laptops.

Chipsets support

Nos encontramos ahora con soporte para prevenir los bugs de varios chipsets. Si reconocemos en alguno de ellos el que tiene nuestra PC, respondemos que sí; en caso contrario, deshabilitamos todos.

Loopback device support

A menos que sepamos lo que estamos haciendo, contestamos que no. Esta característica es necesaria para montar un archivo como un sistema de archivos. Puede servir para chequear un sistema ISO9660 luego de grabar un CD.

RAM disk support

Esta opción nos permite crear un ramdisk en el momento del booteo. Sólo es necesario en caso de crear kernels de instalación o rescate del sistema; respondemos que no.

Parallel port IDE device support

Esta opción sólo es necesaria si contamos con algún dispositivo IDE que se conecte por el puerto paralelo, por ejemplo, algunas lectoras externas de CD-ROM. Respondemos que no.

Networking options

En este apartado, vamos a configurar todo lo relacionado con el soporte de red del kernel (**Figura 6**).

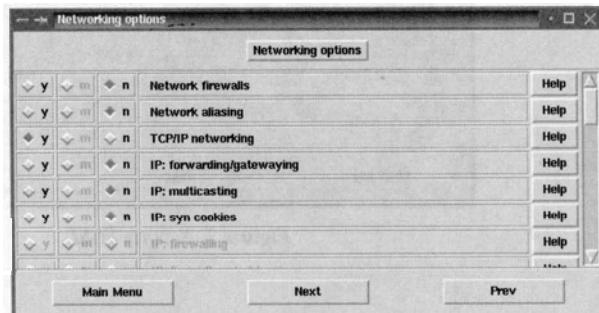


Figura 6. Opciones del kernel para red.

Network firewalls

Un firewall es una PC que actúa como barrera para proteger una subred. Si éste es el destino de nuestra máquina, respondemos que sí. En caso contrario, marcamos no.

Network aliasing

Esto es necesario para asignar múltiples direcciones IP a un mismo dispositivo de red. Respondemos que no.

TCP/IP networking

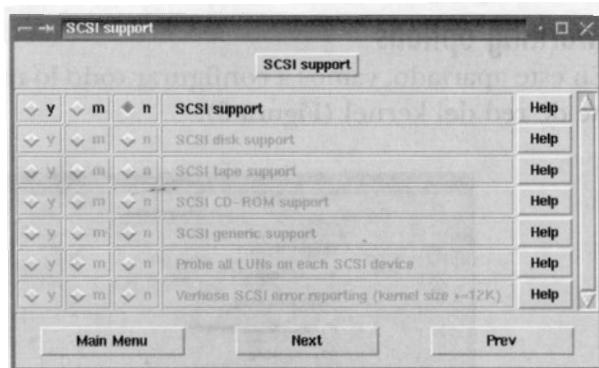
Éste es el protocolo que usará nuestra máquina tanto para la conexión con Internet como con otras máquinas de la red. En cualquiera de los casos, respondemos que sí.

Otras opciones

El resto de las opciones de esta sección son muy específicas y se refieren a TCP/IP o a otros protocolos tales como IPX. Es conveniente dejarlas como se encuentran.

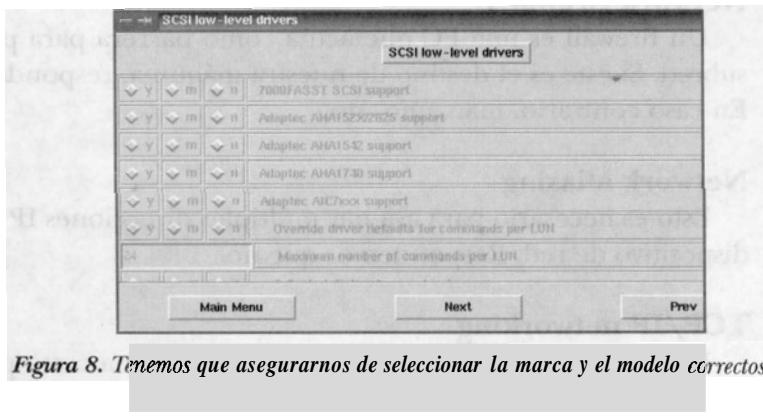
SCSI support

Ésta es la sección dedicada al soporte de dispositivos SCSI. En caso de contar con alguno, seleccionamos cuidadosamente el dispositivo para el que lo necesitamos (**Figura 7**).

**Figura 7.** Soporte SCSI.

SCSI low-level drivers

Aquí seleccionamos la marca y el modelo adecuado de nuestro adaptador SCSI (**Figura 8**).

**Figura 8.** Tenemos que asegurarnos de seleccionar la marca y el modelo correctos.

Network device support

Ahora tenemos que configurar los dispositivos de red. No tiene que ser necesariamente una placa Ethernet; también encontraremos en este apartado la configuración para conexión por cable serie, cable paralelo, módem, etc. (**Figura 9**).

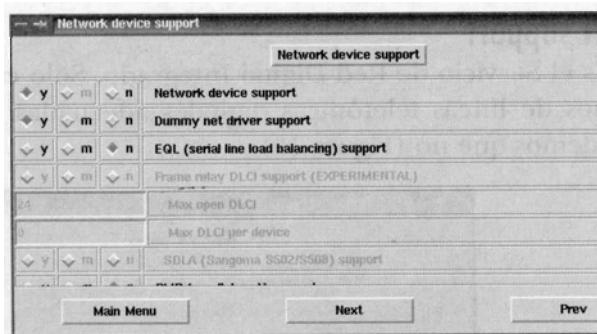


Figura 9. Lo más común es que nuestra placa sea ISA o PCI NE2000 compatible.

Network device support

Respondemos que sí a menos que deseemos aislarnos totalmente del mundo o seamos un naufrago en una isla.

Dummy net driver support

Contestamos que sí en caso de que pensemos usar conexiones PPP o SLIP.

EQL support

Sólo respondemos que sí en caso de contar con dos módems.

PLIP support

Respondemos afirmativamente si vamos a usar el puerto paralelo para conectar otra PC.

PPP support

Éste es el protocolo de conexión punto a punto, una nueva implementación de SLIP que nos sirve, fundamentalmente, para la conexión por puerto serie, módem o una red. Si pensamos conectarnos a Internet, será preciso incluir este soporte.

SLIP support

Obsoleto e innecesario si ya incluimos PPP.

Placas de red

A continuación, se presenta un listado de controladores para todas las placas de red soportadas por el kernel. En caso de no encontrar la que tenemos, una buena opción es incluir soporte para NE2000, ya que la mayoría de las placas usa este protocolo.

ISDN support

Es el Servicio de Red Digital Integrado. Sólo es necesario si disponemos de líneas telefónicas digitales adecuadas. Normalmente, respondemos que no (**Figura 10**)

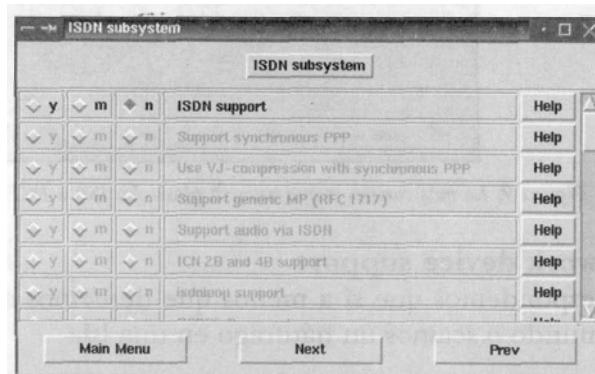


Figura 10. Soporte ISDN.

CD-ROM drivers

Aquí encontraremos soporte para nuestro CD-ROM si no es de la norma SCSI o IDE/ATAPI. Seleccionamos cuidadosamente el que necesitamos, si es necesario (**Figura 11**).

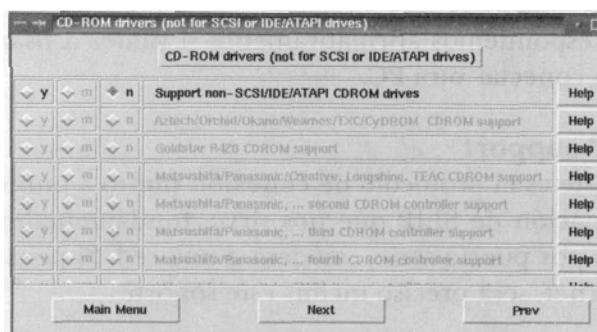


Figura 11. Todos los CD-ROMs con norma propia están soportados.

Filesystems

En este apartado, encontraremos la opciones necesarias para configurar todos los sistemas de archivos a los que queramos tener acceso desde LINUX (**Figura 12**).

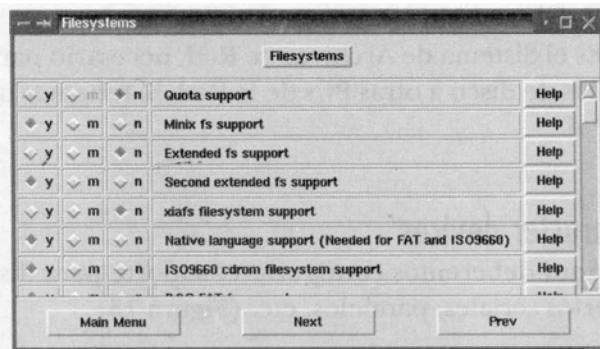


Figura 12. No se deben incluir los sistemas de archivos que no se van a usar; ya que sólo agrandarán el kernel.

Sistemas a incluir

Second extended fs support: propio de LINUX.

Native language support: necesario para FAT e ISO9660.

ISO9660 CDROM filesystem support: soporte para acceder a CD-ROM.

DOS fat support: soporte para disco con FAT DOS.

MSDOS filesystem support: soporte para diquetes DOS.

VFAT fs support: soporte para disco Windows 95+.

Codepage

850: es la necesaria para el lenguaje español.

Set de caracteres

IS08859-1: es el set de caracteres Latin 1 que cubre la mayoría de los idiomas.

Proc filesystem

Es un sistema de archivos virtual que provee información acerca de nuestro sistema; contestamos que sí.

NFS filesystem

Es el Sistema de Archivos de Red, necesario para poder exportar recursos de disco a otras PCs de la Red. Si tenemos una red, contestamos que sí.

Character devices

Aquí deberemos configurar el soporte para dispositivos tales como puertos seriales, paralelos, etc. (**Figura 13**).

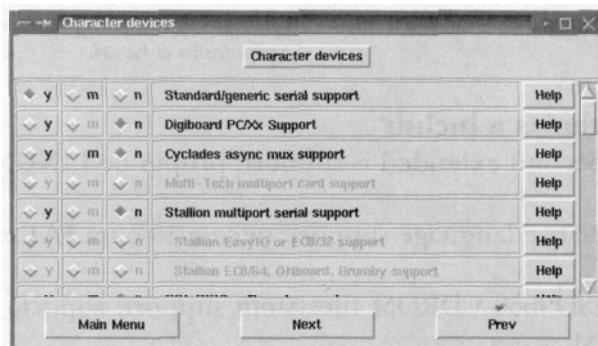


Figura 13. No debemos olvidarnos de incluir soporte para impresora paralelo.

Standard/generic serial support

Soporte genérico para puertos seriales; contestamos que sí.

Parallel printer support

Si tenemos una impresora conectada al puerto paralelo, contestamos que sí.

Mouse support

En caso de que nuestro mouse no sea serial, seleccionamos aquí el controlador adecuado. En las nuevas Pentium II, es necesario elegir PS/2.

Sound

Aquí indicamos la marca, el modelo y la configuración de nuestra placa de sonido (**Figura 14**).

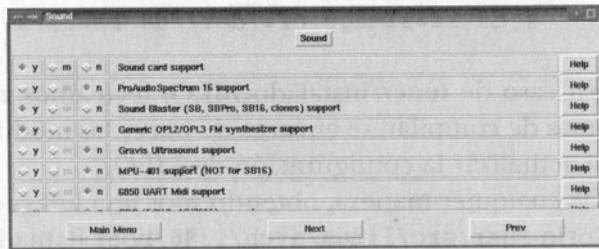


Figura 14. Verificamos el número de puerto e IRQ de nuestra placa.

Kernel hacking

Esta opción es sólo para usuarios avanzados que necesitan información acerca de la actividad del kernel en tiempo de ejecución. No la seleccionamos.

Compilación

Una vez terminada la configuración, vamos a proceder directamente con la compilación.

Ingresamos los siguientes comandos:

```
# make dep;make clean
```

De esta forma, nos aseguramos de que el proceso de creación del núcleo partirá de cero; borrará todos los ficheros creados en compilaciones anteriores, incluso cualquier imagen del núcleo existente (**vmlinux**) y módulos, si es que los hemos incluido en la compilación. Una vez terminado el proceso anterior, introduciremos el comando:

```
# make zImage
```

Este comando compilará la nueva imagen del núcleo. Al final del proceso, se dispondrá de una nueva imagen del núcleo en el directorio **/usr/src/linux/arch/i386/boot**. Esta tarea puede demorar varios minutos, dependiendo del procesador y de la memoria que tenga nuestra PC. Para crear un disco de arranque, podemos teclear directamente **make zdisk** introduciendo previamente un disquete en blanco en la unidad o tecleando al final del proceso de compilación:

```
# dd if=zImage of=/dev/fd0 bs=8192
```

En caso de tener instalado LILO, tecleamos `make zlilo`, lo cual, además de compilar, se encarga de hacer las modificaciones necesarias para actualizar la configuración de LILO.

De cualquier manera, obtendremos una imagen del kernel en el directorio `/usr/src/linux/arch/i386/boot` llamado archivo `zImage`, el cual debemos copiar al directorio `/boot` y al `/usr/src/linux/System.map` con las siguientes instrucciones:

```
#cp /usr/src/linux/arch/i386/boot/zImage /boot  
#cp /usr/src/linux/System.map /boot
```

Lilo

En este apartado, expondremos lo más detalladamente posible el proceso de configuración de este software en un sistema con una partición primaria dedicada a LINUX y otra a DOS, que es la configuración más común. Por supuesto que es recomendable leer la documentación de LILO, ya que aquí trataremos sólo los fundamentos.

El archivo de configuración de LILO se llama `lilo.conf` y está en el directorio `/etc`. Los parámetros que usa son los siguientes:

- **boot=<dispositivo>**: indica el dispositivo (unidad o partición) que contiene el sector de arranque; por defecto, se trata del que está montado como raíz.
- **compact**: reduce el tiempo de carga de LILO y es recomendable cuando lo hemos instalado en disquete.
- **delay=<valor>**: décimas de segundo que el sistema esperará antes de arrancar.
- **disk=<nombre>**: nos permitirá detallar la geometría de ciertos discos duros que LILO no puede manejar normalmente debido al tipo de BIOS que tenemos instalada. Por ejemplo:

```
disk=/dev/hda  
    bios=0x80  
    sectors=32  
    heads=64  
    cylinders=632
```

• **image=<fichero>**: la imagen de LINUX a utilizar puede estar ubicada en una partición (a la que LILO puede tener acceso) o en un disquete; normalmente, utilizaremos la primera opción por defecto.

• **install=<fichero>**: indica qué fichero va a instalarse como sector de arranque; por defecto, se instalará el fichero **/boot/boot.b**.

• **label=<nombre>**: asigna un nombre a la opción en la que se incluye. Junto con **prompt** y **timeout**, esto da la posibilidad de escoger entre varias opciones durante el arranque de la computadora.

• **linear**: algunas BIOS tienen una manera de transformar el par dirección/geometría en los discos que no es compatible con LINUX. Para estos casos, se recomienda utilizar esta opción junto con la geometría de la unidad (opción **disk**).

• **map=<fichero>**: especifica la ubicación del fichero **map** que, por defecto, será **/boot/map**, creado por LILO para contener información crítica del sistema.

• **other=<dispositivo>**: se utiliza junto con el parámetro **table** para que LILO pueda arrancar otros sistemas operativos (DOS en nuestro caso).

• **root=<dispositivo>**: partición que va a ser montada como raíz del sistema de ficheros. De no indicarse, se entiende que es en la que se encuentra LILO en el momento de la instalación (más exactamente la que está configurada en la imagen del núcleo a utilizar y que podemos modificar con el comando **rdev**).

• **table=<dispositivo>**: lugar donde se encuentra la tabla de particiones; información que LILO no maneja y que es necesaria para otros sistemas operativos.

Es conveniente editar el archivo **lilo.conf** y copiar la sección dedicada al kernel preinstalado, modificando los datos necesarios para bootejar con el nuevo núcleo. Estos cambios son:

• **Image**: donde debemos poner el nombre de la nueva imagen del kernel.

• **Label**: la etiqueta que nos mostrará LILO en el momento de arranque para seleccionar la nueva opción.

• **root**: es el dispositivo que se montará como raíz.

Finalmente, debemos ejecutar **#lilo** para que se actualice la tabla de arranque de LILO.

Cuando arranquemos nuevamente nuestra PC y se nos presente **LILO**, debemos pulsar TAB para ver los kernels que podemos usar para bootear.

REDES EN LINUX

| | |
|---------------------------|-----------|
| Introducción | .Pág. 139 |
| TCP/IP | .Pág. 140 |
| Capas | .Pág. 140 |
| Kernel | .Pág. 142 |
| Configuración | .Pág. 143 |
| Netcfg | .Pág. 144 |
| Names | .Pág. 144 |
| Hosts | .Pág. 144 |
| Interfaces | .Pág. 145 |
| Servidor de nombres | .Pág. 147 |

"El progreso consiste en el cambio."

Miguel de Unamuno

"Aquel que duda y no investiga se torna no sólo infeliz, sino también injusto."

Blas Pascal

Introducción

Si lo que necesitamos es un sistema operativo para armar una red, LINUX es la mejor alternativa. Es fácil de configurar, totalmente confiable, y muy robusto y estable. En la actualidad, LINUX soporta:

- La mayoría de las tarjetas de tipo Ethernet existentes.
- Protocolos de transmisión vía el puerto serie SLIP (*Serial Line IP*) y PPP (*Point to Point Protocol*).
- Compresión de encabezamiento de paquetes con el algoritmo Van Jacobson.
- PLIP para la conexión de PCs vía el puerto paralelo.
- EQL *Load Balancing*, que permite el uso de más de una conexión simultánea con otra máquina, con lo que se multiplica el ancho de banda disponible.
- NFS: los sistemas de ficheros remotos son accesibles vía este protocolo, lo cual posibilita el uso transparente para el usuario.
- IPX: protocolo utilizado por las redes de tipo Novel.
- NIS (*Netwk Information System*): protocolo creado por la empresa SUN destinado a poner cierta información vital a todas las máquinas de una red y cuyo objetivo es facilitar su manejo.
- ARCnet.
- Redes de tipo *Token Ring* (IBM).

- Appletalk: para compartir ficheros y dispositivos con máquinas Macintosh.
- Redes sin cable: generalmente vía dispositivos infrarrojos.
- ISDN (o RDSI, Red Digital de Servicios Integrados).
- ATM: protocolo de comunicaciones.
- IP *firewalling*: para configurar la máquina LINUX como cortafuegos de una red.
- IP *Accounting*: contabilidad del sistema.
- IP *tunnelling*: permite el encapsulamiento de un protocolo dentro de otro ('disfrazar' un protocolo para que tenga apariencia de otro). También se utiliza para el caso de IP 'móviles', en el que una máquina tiene una dirección IP fija pero no siempre está ubicada en un mismo lugar.

TCP/IP

TCP/IP (*Transport Control Protocol / Internet Protocol*) es el protocolo por defecto de LINUX, el estándar en todos los UNIX y también el estándar de Internet. Sus orígenes se remontan al año 1969, en la Agencia de Proyectos Avanzados de Investigación en Defensa de los Estados Unidos. Allí se inició la legendaria ARPANET, padre de Internet. En el año 1983, Internet adopta como protocolo estándar el TCP/IP, que luego también sería incorporado como el estándar en los sistemas UNIX.

Capas

La transmisión de datos en TCP/IP se hace a través de paquetes. Los paquetes son fragmentos de datos cubiertos por capas, cada una de las cuales corresponde a un protocolo. Podemos imaginarnos a cada capa como una caja de embalaje; así, enviamos un datagrama (paquete de datos) desde nuestra PC a otra de la misma red a través de una placa Ethernet.

Como podemos ver en la **Figura 1**, cada uno de los embalajes corresponde a uno de los protocolos que el datagrama debe atravesar para llegar a su destino. Primero se recubren los datos con el *header* del protocolo TCP, luego con el de IP y, finalmente, con el de la placa de red Ethernet. Cuando este datagrama llega a su destino, el receptor debe desembalarlo en el orden inverso al que fue empaquetado.

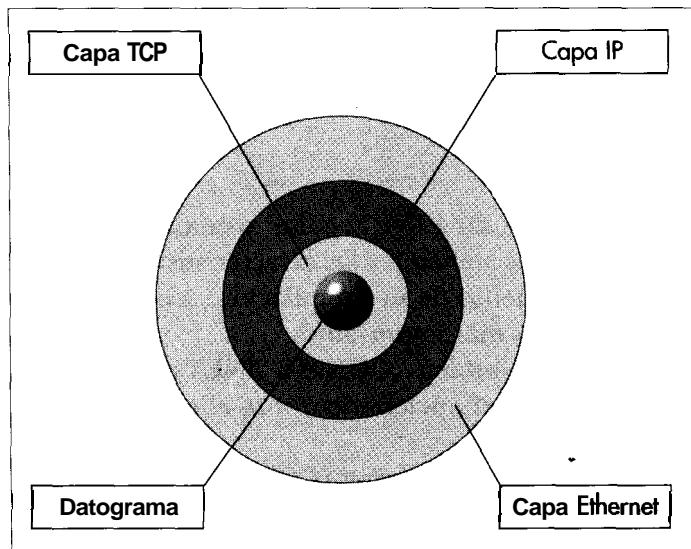


Figura 1. Esquema de un datagrama y sus "envoltorios".

Técnicamente, sucede lo siguiente. El protocolo Ethernet compilado en el kernel primero consulta si la línea de transmisión está ocupada. Si es así, espera un momento e intenta nuevamente este procedimiento, que se llama "Detección de colisión". Cuando la línea está disponible, la PC de origen recubre los datos con los *headers* correspondientes, es decir con el de TCP, IP y Ethernet, en ese orden, y los envía.

El kernel de la máquina que recibe el paquete le quita el primer recubrimiento y luego procede a interpretar los datos mediante IP. Éste es un protocolo que no garantiza la transmisión de los datos, porque no existe ningún sistema de verificación a este nivel que asegure que éstos han sido recibidos.

En este protocolo nos encontramos con el concepto de número de IP, un sistema de clasificaciones para nodos de una red mediante la utilización de cuatro octetos. Los 32 bits de los cuatro octetos se pueden usar de tres maneras distintas. A continuación, podemos ver los tipos de red que actualmente tenemos en IP.

| Clase | Red | Nodo | Descripción |
|-------|---------|---------|---------------------------------------|
| I | 8 bits | 24 bits | Un octeto para red y tres para nodo |
| II | 16 bits | 16 bits | Dos octetos para red y dos para nodo |
| III | 24 bits | 8 bits | Tres octetos para red y uno para nodo |

Tipos de red en IP.

Kernel

Para poder usar LINUX como sistema operativo de red, debemos prestar especial atención a las características que incluiremos en el kernel en el momento de su compilación. A continuación, una breve descripción de las más importantes.

- **TCP/IP networking (CONFIG_INET):** contestar afirmativamente, de forma que el núcleo incorpore el código necesario para manejar este protocolo.
- **IP forwarding/gatewaying (CONFIG_IP_FORWARD):** debemos habilitar esta opción si LINUX va a realizar las tareas de pasarela (enrutado de paquetes).
- **IP firewalling (CONFIG_IP_FIREWALL):** sólo debemos incluir esta opción en el caso de que nuestra PC actúe como un "cortafuegos" o firewall.
- **IP accounting (CONFIG_IP_ACCT):** para el caso de que necesitemos medir el uso de recursos de la red.
- **PC/TCP compatibility mode (CONFIG_INET_PCTCP):** evita errores cuando LINUX ha de soportar máquinas que utilizan este protocolo.
- **Reverse ARP (CONFIG_INET_RARP):** soporte del protocolo RARP (*Reverse Address Resolution Protocol*), utilizado para la asignación de dirección IP a terminales sin disco (terminales 'bobas').
- **Assume subnets are local (CONFIG_INET_SNARL):** influye en el tamaño de los paquetes a transmitir, dependiendo de si sólo asume como local a las máquinas directamente conectadas a la red o también a otras más 'lejanas'.
- **Disable Nagle algorithm (normally enabled):** trata de evitar el envío de paquetes muy pequeños por la red.
- **The IPX protocol (CONFIG_IPX):** protocolo de redes de tipo Novell.

Configuración

La configuración de una red en LINUX es sencilla. Lo único que necesitaremos es un editor de texto para introducir las líneas adecuadas en los archivos de configuración, todos los cuales tienen una configuración por defecto y están ampliamente documentados para poder interpretar cada una de las líneas.

Todos los archivos de configuración se encuentran, como debe ser, en los directorios **/etc**. Veámoslos uno por uno y analicemos su función y contenido.

HOSTNAME: contiene una sola línea con el nombre de nuestra máquina.

hosts: aquí incluiremos los nombres y los números de IP de las máquinas que forman nuestra red. Este archivo es consultado en primera instancia para la resolución de nombres. Como mínimo, debemos tener una línea para el loopback 127.0.0.1 **localhost**.

hosts.conf: indica a LINUX la forma en que ha de proceder a la hora de resolver nombres de las máquinas de la red.

resolv.conf: contiene las direcciones IP de los servidores de nombres de dominio. También permite indicar el dominio de la máquina en caso de tener uno asignado.

inetd.conf: en este archivo, tenemos la configuración del demonio de Internet, que está a la espera de peticiones de conexión. En el caso de que llegue una, dispara la ejecución del servicio adecuado consultando en el archivo **/etc/services** el número y protocolo a utilizar para cada servicio.

services: listado del nombre, número y protocolo asignado a cada *socket*.

protocol: contiene todos los protocolos que podemos utilizar con LINUX y el número asignado a cada uno de ellos.

Además, existen dos archivos muy importantes para la inicialización del software implicado en la gestión de redes y en la declaración de las variables necesarias: **rc.inet1** y **rc.inet2**.

En la distribución Red Hat, también contamos con una poderosa herramienta de configuración para redes, que se encuentra en el Panel de Control (Control-panel). Su uso es muy sencillo e intuitivo y nos permite configurar la red sin mayores complicaciones.

Netcfg

RedHat LINUX cuenta con una poderosa herramienta de configuración para redes: **Netcfg**. Para ejecutarla, podemos hacerlo desde Control Panel o invocarla con **netcfg** desde una terminal. En la **Figura 2**, podemos ver la pantalla principal.

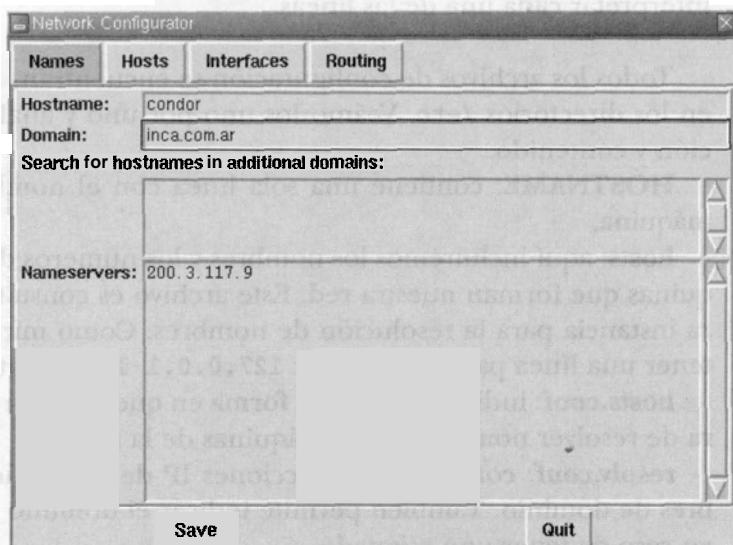


Figura 2. Definición del Hostname y del nombre de dominio de nuestra red.

Names

Aquí tenemos que colocar el nombre de nuestra PC y el nombre de la red. En el ejemplo, el hostname es **condor** y la red es **inca.com.ar**. En nameserver, debemos poner el número IP del servidor de nombres de dominio; en el ejemplo, 200.3.117.9.

Hosts

En esta sección, tenemos que declarar las PCs de nuestra red. Por lo menos, debe estar declarado el *loopback*, como se ve en la **Figura 3**.

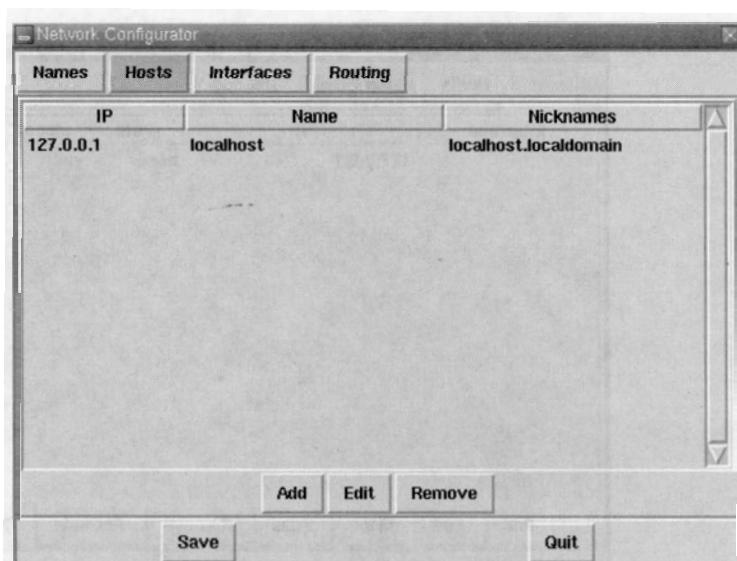


Figura 3. Declaración de las PCs conectadas a nuestra red.

Para agregar nuevas máquinas a la red, es preciso pulsar en el botón Add, con lo que aparecerá una nueva ventana solicitándonos el número de IP, el hostname y un alias para la nueva máquina (**Figura 4**).

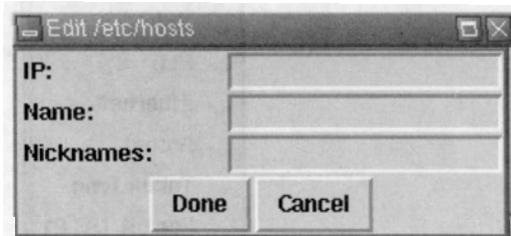


Figura 4. Datos necesarios para agregar una terminal a nuestra red.

Interfaces

En esta sección, configuraremos las interfaces que usaremos para conectar nuestra red (**Figura 5**). Tenemos también el botón necesario para agregar interfaces; una vez que lo pulsamos, aparecerá una nueva ventana (**Figura 6**), donde deberemos seleccionar el protocolo adecuado de acuerdo con el dispositivo que elijamos.

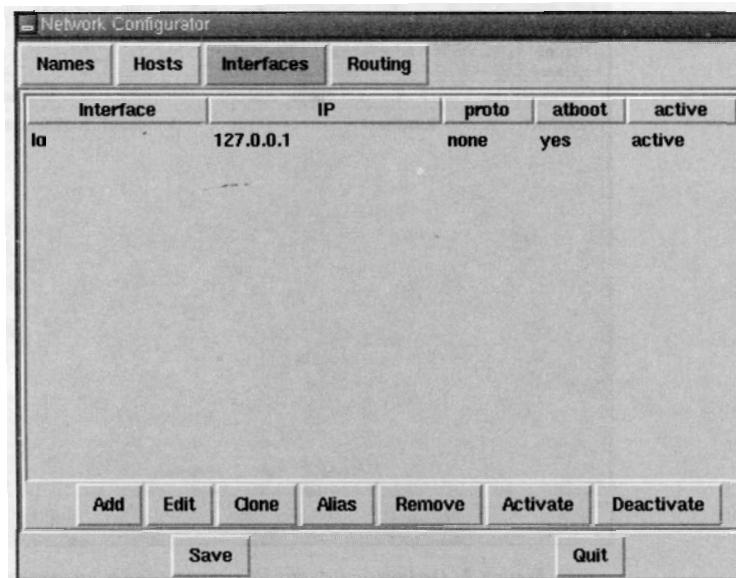


Figura 5. Configuración de las interfaces de Red.

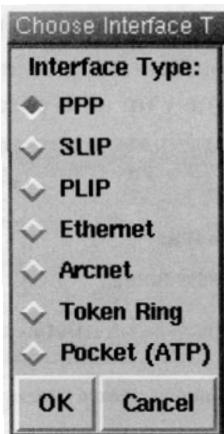


Figura 6. Selección del protocolo para la interfase.

Finalmente, en *routing* están las opciones necesarias para configurar nuestra PC como pasarela y para rutear los datagramas (**Figura 7**).

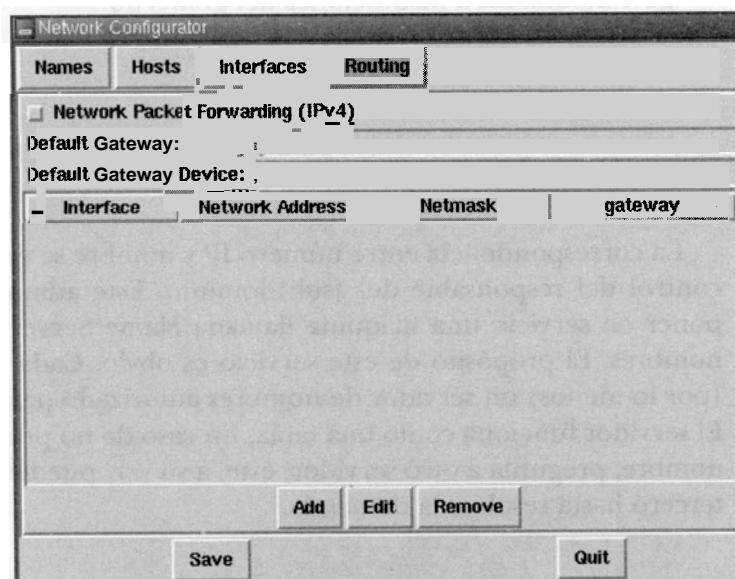


Figura 7. Configuración para el Gateway.

Servidor de nombres

Debido a la gran cantidad de nodos con que cuenta Internet, fue necesario crear un sistema de denominación más accesible para los humanos. Este sistema se basa en dos premisas fundamentales:

- Carácter mnemotécnico
- Descentralización de la administración de la red.

Así surgió el sistema de Nombre de Dominio. Los dominios se sub-

.gov: destinados a instituciones del gobierno.

.ml: para organizaciones militares.

A su vez, existen dominios para determinar el país donde reside el nodo, por ejemplo .ar, .bo, .br, .cl, etc.

La correspondencia entre número IP y nombre se establece bajo el control del responsable del (sub)dominio. Este administrador debe poner en servicio una máquina llamada Name Server o servidor de nombres. El propósito de este servicio es obvio. Cada dominio tiene (por lo menos) un servidor de nombres autorizado para ese dominio. El servidor funciona como una tabla. En caso de no poder resolver un nombre, pregunta a otro servidor; éste, a su vez, puede consultar a un tercero hasta resolver la demanda.

XWINDOW

| | |
|---------------------------------------|----------|
| Definición | Pág. 151 |
| Conceptos | Pág. 152 |
| Servidores X | Pág. 152 |
| Manejadores de ventanas | Pág. 153 |
| Comparaciones | Pág. 154 |
| FVWM-95 | Pág. 154 |
| AfterStep | Pág. 155 |
| Enlightenment | Pág. 156 |
| Window Maker | Pág. 157 |
| KDE | Pág. 158 |
| BlackBox | Pág. 159 |
| Emulador de terminal | Pág. 160 |
| Configuración de XWindow | Pág. 161 |

*"Si así fue, así pudo ser; si así fuera, así podría ser; pero como no es, no es.
Eso es lógica."*

Lewis Carroll

"Si hay victoria en vencer: al enemigo, la hay mayor cuando el hombre se vence a si mismo. "

José de San Martín

Definición

El sistema XWindow, también conocido como X, es una interfase para el usuario orientada al uso de teclado, mouse y una pantalla gráfica.

Todo sistema de ventanas es una colección de programas que le permiten utilizar y mostrar varias aplicaciones en una única pantalla. Esto se logra dividiendo la pantalla en áreas más pequeñas (ventanas), con la posibilidad de que éstas se superpongan y se puedan ejecutar varias sesiones en forma simultánea.

La mayoría de los sistemas de ventanas utilizan una abstracción llamada "escritorio" (desktop); de esta forma, la pantalla es tratada como si fuera la parte superior de un escritorio, y cada ventana de programa como si fuera un pedazo de papel. En un escritorio real, podemos poner hojas una al lado de otra, apilarlas unas sobre otras, moverlas, doblarlas y muchas posibilidades más. De manera similar, un sistema de ventanas como X nos permite colocar ventanas en la pantalla, apilarlas, colocarlas en forma contigua, moverlas, etc. Este modelo fue inventado por Apple para sus computadoras Mac, y luego ampliamente copiado a todas las plataformas.

En LINUX, XWindow nos provee un interfase completamente gráfica, con menús desplegables y un uso intuitivo del mouse. También mantiene los conceptos de ventanas, íconos y menús colgantes, justamente como en Macintosh. Con este sistema, es posible mover ventanas, cambiarles el tamaño (resize), transformarlas en íconos, etc.

Como seguramente la mayoría de los lectores estarán acostumbrados al uso de Windows 98, la forma menos traumática de migrar a LINUX será usando estas Interfases Gráficas para el Usuario (GUI).

La implementación del sistema de ventanas de LINUX consta de dos partes fundamentales, servidores y clientes, que veremos más adelante.

Conceptos

Aquí se definen algunos conceptos que son propios de los sistemas de ventanas.

Aplicación: son los programas que se usan en el sistema X, por ejemplo, graficadores, navegadores, juegos, editores de textos, etc.

Cursor: es un puntero en la pantalla, por lo general un cuadrado o línea que indica dónde irá el primer carácter cuando se comience a escribir.

Display: es la pantalla de su computadora, el monitor.

Ícono: dibujo simbólico que representa una aplicación. Para transformar un ícono en una aplicación, se debe apuntar con el cursor del mouse sobre él y pulsar el botón.

Mouse o ratón: dispositivo mecánico con botones que sirve para apuntar.

Cursor del mouse: muy similar al cursor regular que definimos previamente. Cuando se mueve el mouse, el cursor se mueve en la pantalla para indicar dónde apunta. Es común que la forma del cursor varíe dependiendo de su posición y función.

Ventana: área limitada en la que una aplicación realiza la mayoría de sus funciones de entradas y salidas.

Servidores X

Por una parte, la implementación de ventanas se hace sobre un servidor gráfico, llamado XFREE86, desarrollado en el Instituto Tecnológico de Massachusetts (MIT). Se habla de servidor gráfico porque, obviamente, está desarrollado usando la arquitectura cliente-servidor. De esta manera, X es un servidor, y Netscape, Eterm, GIMP y todos los programas que corren bajo X son clientes. También podemos ejecutar desde una terminal cualquier software que no esté específicamente diseñado para X. Existen versiones comerciales de servidores gráficos, pero como el espíritu de LINUX es el software libre, sólo comentaremos los de libre distribución. Veamos a continuación cuáles son las funciones principales de un servidor X:

- Administración de bajo nivel de ventanas (creación y destrucción).
- Gestión de recursos del sistema.
- Dibujo de los gráficos necesarios para crear la interfaces.

- Gestión y control de eventos (pulsado de una tecla, mouse, etc.).
- Control de errores.
- Control de la pantalla.
- Interconexión entre programas (cortar y pegar, etc.).

Para cada placa de video, existe un servidor específico, aunque muchas de ellas se incluyen en servidores genéricos, como el SVGA.

La versión actual del entorno XWindow es la número 11, en su revisión número 6, y pertenece al denominado *X Consortium*, grupo formado en 1988 por el MIT junto a otras empresas, como DEC e IBM, cuyo objetivo es la cooperación con la industria para la creación y evolución de un estándar gráfico basado en XWindow.



HAY QUE SABERLO

La versión del software de XFREE86 al momento de terminar este libro es la 3.3.3-1

Manejadores de ventanas

En una segunda etapa, tenemos que seleccionar el *Window Manager* o manejador de ventanas, que es la interfase real con el usuario. Éste también es un cliente para X, de los que existe un gran variedad, con apariencias totalmente diferentes. Algunos son clásicos y muy completos, como KDE, y otros están basados en diferentes sistemas operativos, como FVWM95, que es una réplica de Windows de Microsoft, o AfterStep y Window Maker, basados en NextStep de las computadoras Next. También tenemos algunos totalmente inéditos y originales, como Enlightenment o BlackBox.

Las funciones principales del Manejador de ventanas son:

- Redimensionar las ventanas.
- Mover ventanas.
- Iconizar y desiconizar ventanas (convertir en íconos y viceversa).
- Crear barras de títulos y desplazamiento, bordes, botones, etc.
- Controlar las ventanas.
- Lanzar aplicaciones.

Comparaciones

Veamos ahora una breve descripción de algunos de los manejadores de ventanas más usados y una tabla en la que se comparan sus características. Resulta un tanto difícil hacer esta comparación, dado que no es sencillo clasificar algunos de los factores involucrados, tales como la posibilidad de personalización, documentación, estabilidad y facilidad de instalación. Por supuesto, la siguiente tabla es un listado de características soportadas que no debe tomarse como una calificación de calidad.

| Manejador de ventanas | Pantalla virtual | Varios escritorios | Soporte XPM | Menús fijables | Manejo de sesiones |
|-----------------------|------------------|--------------------|-------------|----------------|--------------------|
| FVWM-95 | ✓ | ✓ | ✓ | ✗ | ✗ |
| AfterStep | ✓ | ✓ | ✓ | ✗ | ✗ |
| Enlightenment | ✓ | ✓ | ✓ | ✓ | ✗ |
| Window Maker | ✗ | ✓ | ✓ | ✓ | ✗ |
| KDE | ✓ | ✓ | ✓ | ✓ | ✓ |
| BlackBox | ✓ | ✓ | ✓ | ✗ | ✗ |

Manejadores de ventanas y sus características.

FVWM95



Logo oficial de FVWM95.

Fue uno de los más usados en un principio, aunque ahora lo utiliza muy poca gente. Es completo y amigable; para quien viene de Windows es una buena alternativa, pero cuando el lector conozca los otros, difícilmente opte por seguir usando éste. La **Figura 1** nos permite ver la apariencia que presenta.



Figura 1. Aspecto de FVWM95.

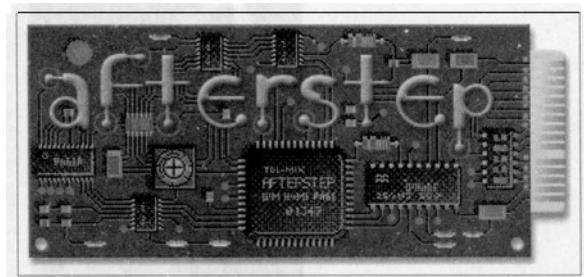


WWW

Sitio oficial de FVWM: [Ftp://mitac11.uia.ac.be/html-test/fvwm95.html](ftp://mitac11.uia.ac.be/html-test/fvwm95.html).

MÁS DATOS

AfterStep



Logo oficial de AfterStep.

El paso siguiente es un clon de NextStep de las computadoras NEXT, el sucesor de Apple. Es un GUI muy estable y de apariencia agradable, que si bien no lo usan muchas personas, es una buena opción. Un aspecto desfavorable es que hace mucho tiempo que no sale una nueva versión (**Figura 2**).

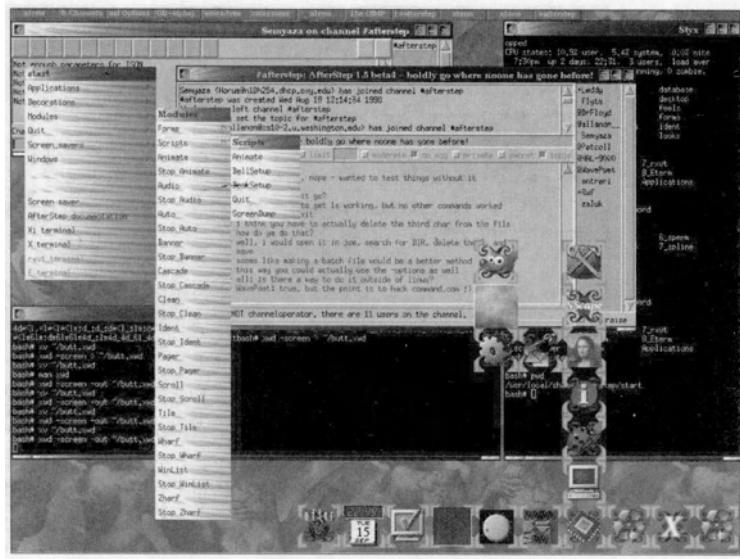


Figura 2. Pantalla capturada de AfterStep

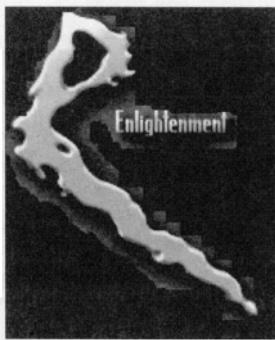


WWW

Sitio oficial de AfterStep: www.afterstep.org.

MÁS DATOS

Enlightenment



Logo oficial de Enlightenment

Si tuviera que definir E con una sola palabra, ésa sería "original". E es un manejador de ventanas que rompe todos los esquemas. Es totalmente configurable, es decir que se puede modificar hasta el más mínimo detalle. Lamentablemente, todavía es muy inestable. Aun así, es

el preferido de una gran mayoría de usuarios. Originalmente de Australia, en la actualidad es parte de los desarrollos de RedHat Labs, donde Rasterman –uno de sus creadores, junto a Mandraque– trabaja en la integración de E con los GTK themes de CVS. La **Figura 3** muestra una de las tantas posibles apariencias de Enlightenment.



Figura 3. Pantalla de Enlightenment.



WWW

Sitio oficial de Enlightenment: www.rasterman.com.

MÁS DATOS

Window Maker



Logo oficial de Window Maker

Otro intento bien logrado de imitar a NextStep. Uno de los aspectos más atractivos es la posibilidad de aplicarle temas de escritorio. Es bastante estable y de apariencia agradable. No sobresale por su origi-

nalidad, pero es una muy buena opción para quien necesita un WM no muy grande y de buenas prestaciones. En la **Figura 4** se observa un ejemplo de pantalla donde se está recompilando el kernel.

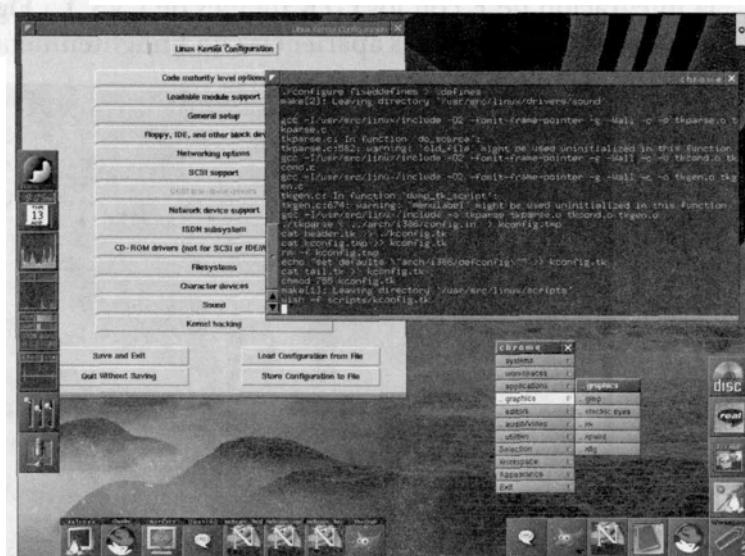


Figura 4. Pantalla de Window Maker.

i **LINUX** MÁS DATOS

Sitio oficial de Window Maker: www.windowmaker.org.

KDE



Logo de KDE

KDE es, tal vez, el más completo de todos los manejadores y el que recomiendo tanto para el que recién empieza como para el veterano. No sólo es un manejador de ventanas, sino un completo set de herramientas de configuración y programas de soporte tales como juegos, programas para redes, utilidades, emulador de terminal, etc. (**Figura 5**).

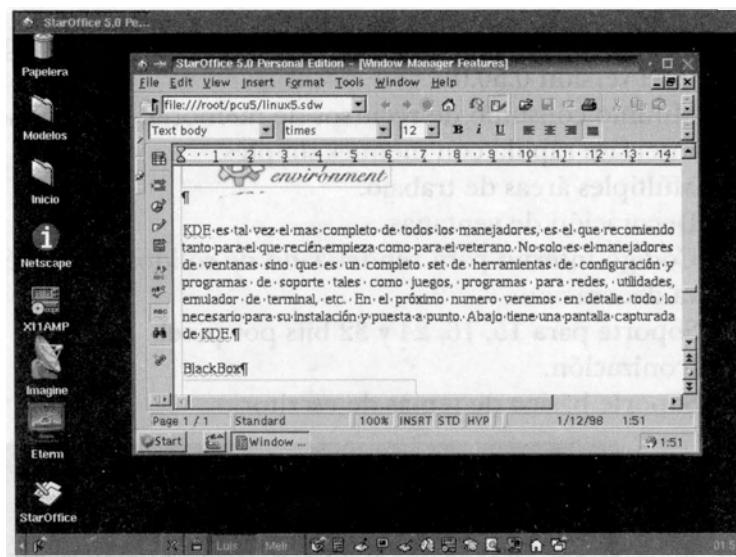


Figura 5. Pantalla de KDE.



WWW

Sitio oficial de KDE: www.kde.org.

MÁS DATOS

BlackBox



Logo oficial de BlackBox

BlackBox es mi manejador preferido, no por ser completo ni por su aspecto, ya que debo reconocer que es el más simple y modesto de todos (**Figura 6**). Cumple satisfactoriamente con las tareas mínimas y consta sólo de dos ejecutables y cuatro archivos de configuración. A continuación, presentamos sus características principales.

- Código fuente pequeño, 10.914 líneas totales de programa en su última versión 0.50.0.
- Mínimo consumo de recursos de memoria, CPU y Xserver.
- Interfase rápida con menúes simples.
- Múltiples áreas de trabajo.
- Decoración de ventanas.
- Código interno para renderizado de sólidos, degradés y relieves para decoración.
- Soporte para 15, 16, 24 y 32 bits por pixel.
- Iconización.
- Soporte básico de temas de escritorio.

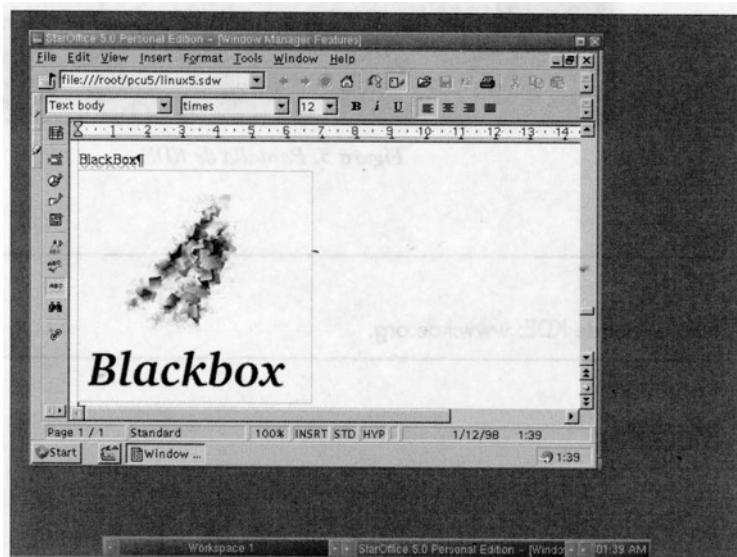


Figura 6. Pantalla de BlackBox.

i **LINUX**

MÁS DATOS

Sitio oficial de BlackBox: blackbox.wiw.org.

Emulador de terminal

Un emulador de terminal es un programa que actúa como si fuera una terminal, enviando y recibiendo caracteres de la computadora. El emulador de terminal por defecto de X11 es **xterm**, aunque existen

muchos otros, como Eterm, Aterm, rxvt, etc. Emula un terminal DEC VT-102; por defecto, xterm utiliza una ventana de 24 líneas por 80 caracteres y el tamaño de una pantalla de un VT-102 real.

Configuración de XWindow

Para configurar Xwindow, contamos con varias herramientas. Vamos a usar la más común y difundida de todas, **xf86config**. Para ejecutarla, sólo debemos invocarla desde la línea de comandos y este programa generará el archivo de configuración que se localiza en el directorio **/etc/X11**. Está estructurado en siete secciones principales, cada una de las cuales contiene información referente a distintos parámetros que el servidor gráfico necesitará para funcionar; éstas son:

- **Files:** directorios donde va a buscar los ficheros necesarios para poder utilizar distintas fuentes y colores.
- **ServerFlags:** algunas opciones a utilizar a la hora de ejecutar el servidor X.
- **Keyboard:** configuración de algunas teclas especiales y la forma en que el servidor responderá ante ellas.
- Pointer: para la configuración del mouse.
- Monitor: datos sobre el monitor.
- Device: aquí es donde especificaremos todos los datos referentes a la tarjeta gráfica.
- Screen: configuración del área de trabajo de la que podemos disponer en nuestra pantalla.

Todas las secciones tienen una sintaxis similar:

Section "Nombre de la Sección"

```
Parámetros  
EndSection
```

SEGURIDAD

| | |
|---|----------|
| Introducción | Pág. 165 |
| Planteo | Pág. 165 |
| Evaluación de riesgos | Pág. 166 |
| Evaluación de costos | Pág. 166 |
| Estrategia de protección | Pág. 167 |
| Análisis | Pág. 167 |
| Evaluación de riesgos | Pág. 167 |
| Intrusos | Pág. 168 |
| Tipos de ataque | Pág. 169 |
| Ataques de software | Pág. 170 |
| Valor intrínseco y derivado de las áreas protegidas | Pág. 171 |
| Servidores de Internet | Pág. 172 |
| Definición de política y estrategia | Pág. 172 |
| Procedimiento para la detección de intrusos | Pág. 172 |
| Servidores | Pág. 174 |
| Medidas | Pág. 174 |

"Muchos años después, frente al pelotón de fusilamiento, el coronel Aureliano Buendía había de recordar aquella tarde remota en que su padre lo llevó a conocer el hielo."

Cien años de soledad, Gabriel García Márquez

Introducción

A fin de elaborar un plan rector de seguridad, se lo debe encarar desde tres puntos de vista:

- **Planteo**
- **Análisis**
- **Definición**

En el planteo, se pretende establecer el método elegido para realizar el posterior análisis y la elaboración de las propuestas. El análisis apunta a realizar un estudio de situación y relevamiento del estado actual del sistema. Por su parte, la definición es la propuesta de políticas y estrategias a fin de establecer un sistema de seguridad que garantice tanto la integridad de los datos como la constancia del servicio.

Planteo

El estado actual de desarrollo de las comunicaciones y del uso de computadoras hace imprescindible la necesidad de encarar la problemática de la seguridad, tanto en lo que se refiere a la integridad de los sistemas como a la información almacenada en ellos. Los datos tienen una importancia cuantificable no sólo a partir de su valor propio o intrínseco, sino también del tiempo y trabajo que demandó elaborarlos.

Para la definición de una política de seguridad, es necesario considerar tres puntos fundamentales.

1. **Evaluación de riesgos**
2. **Evaluación de costos**
3. **Estrategia de protección**

Evaluación de riesgos

Se deberá minimizar el riesgo tanto de pérdida o robo de información como de interrupción de los servicios. Para ello, se debe hacer una valoración de riesgos, considerando los siguientes ítem:

- El valor de lo que queremos proteger nunca debe ser mayor que el costo de la protección ni que el costo en que deberá incurrir el posible intruso para acceder a los datos protegidos.
- El tipo de intruso y con qué nivel de conocimientos y recursos cuenta, como así también su edad, condición social y lugar de residencia a fin de iniciar las posibles acciones legales.
- El valor intrínseco de lo que queremos proteger, es decir cuánto nos cuestan los elementos factibles de ser afectados, por ejemplo, el valor del software, el hardware, la contabilidad, los productos en desarrollo, la organización, la sistematización, etc.
- El valor derivado de la pérdida, es decir, lo que puede llegar a costar recuperar o rehacer lo perdido o dañado, por ejemplo, compra de software o hardware dañado, reorganización y resistematización, etc.

Evaluación de costos

En este aspecto se contempla la inversión mínima que se deberá hacer, tanto en equipamiento y software como en recursos humanos.

Para la evaluación de los costos es preciso contemplar:

- Personal afectado específicamente e indirectamente a la protección.
- Hardware y software necesarios para las medidas de seguridad.
- Tiempo dedicado al mantenimiento.
- Por sobre todas la cosas, se debe recordar siempre que el valor de los elementos a proteger debe ser superior que el valor de la protección y menor que los recursos que se deban emplear para acceder a esos elementos.

Estrategia de protección

Para plantear una estrategia de protección eficiente, es necesario considerar varios aspectos a fin de no dejar descubierto ningún flanco ni que quede alguna posible inseguridad sin contemplar:

- **Físico:** aquí consideraremos los recaudos que se deberán tomar para que todo el equipamiento del sistema permanezca lo suficientemente aislado y protegido desde el punto de vista físico.
- **Lógico o digital:** en esta sección se deben tener en cuenta las medidas que asegurarán la integridad y protección del sistema de accesos y ataques a través de medios digitales.
- **Humano:** es importante evaluar los diferentes niveles de acceso que tiene el personal dependiente a las áreas a proteger.
- **Logístico:** el planteo de un esquema y política de protección, como así también los protocolos y rutinas de control deben ser lo suficientemente desarrollados y estrictos como para garantizar un nivel aceptable de seguridad, pero también deben ser lo necesariamente flexibles como para no causar molestias ni mal servicio a los usuarios.

Análisis

Evaluación de riesgos

Plataforma instalada

Lo primero que se considera en la evaluación de riesgos es el valor de lo que se desea proteger. A tal fin, enumeramos a continuación todos los ámbitos que son de riesgo y que se deberán incluir en las medidas de seguridad.

- *Servidores*
- *Dispositivos*
- *Intranets*

Servidores

El factor de riesgo principal en este ámbito es la información personal de los usuarios, no sólo por el derecho a la privacidad de las per-

sonas, sino también por el peligro de posibles demandas contra el administrador del sistema por robo o modificación de la información almacenada temporalmente en los servidores.

El segundo lugar en cuanto a prioridad lo ocupa la interrupción del servicio, un aspecto fundamental para la buena imagen y calidad del servicio prestado.

Finalmente, tenemos que cubrir la seguridad del sistema contra el robo de cuentas, la creación de cuentas falsas y los ataques de denegación de servicio a abonados.

Intrusos

En el análisis de los potenciales intrusos, es preciso considerar como mínimo los siguientes perfiles y tipos:

- Usuarios no técnicos motivados por la curiosidad.
- Personal calificado (estudiantes, programadores, técnicos, etc.), que considera un reto personal violar la seguridad de un sistema.
- Delincuentes en intentos de obtener beneficios económicos.
- Espionaje comercial o militar.

Obviamente, el esfuerzo dedicado a la seguridad depende del tipo de intrusión que se pretenda prevenir.

Podemos definir tres requisitos necesarios para la seguridad dentro de un sistema informático:

Privacidad: la información debe ser accesible para lectura únicamente por las partes autorizadas. Este tipo de acceso incluye la impresión, tanto por pantalla como por impresora, e incluso la revelación de la existencia de un determinado objeto o suceso.

Integridad: los elementos relacionados con la seguridad únicamente pueden ser modificados por las partes autorizadas. La modificación incluye escritura, creación, cambio de estado y borrado.

Disponibilidad: los elementos relacionados con la seguridad deben estar disponibles sólo para las partes autorizadas.

Intrusos externos

- Hackers: el hacker es, por lo general, una persona joven que se encuentra cursando estudios universitarios o está recién recibido, aunque esta condición no es excluyente. No representa más riesgo que el de poner en evidencia la vulnerabilidad del sistema, aunque no son dañinos por naturaleza.
- Crackers: son el principal factor a considerar como posibles intrusos y atacantes. Responden a un perfil más amplio que el del hacker, ya que pueden ser adolescentes o adultos con grandes recursos económicos y equipamiento. Deben ser catalogados como delincuentes, ya que, a diferencia de los hackers, éstos lucran con lo robado, ya sea en beneficio personal, realizando un chantaje o actuando como mercenarios para terceros.

Intrusos internos

- Empleados: constituyen otro factor de riesgo a considerar, no sólo por la posibilidad de tener empleados deshonestos en la empresa, sino por el riesgo que implican actitudes revanchistas y de venganza ante posibles sanciones o medidas disciplinarias.
- Allegados: aunque resulte odioso, también se debe considerar el riesgo que representan personas que, por una razón u otra, tienen acceso al sistema, tanto a través de medios digitales como físicos (por ejemplo, quienes tiene acceso al sistema con cuentas con shell y quienes pueden acceder físicamente a las terminales).

Tipos de ataque

Dependiendo de la metodología utilizada, podemos identificar cuatro tipos básicos de ataques:

- Interrupción: se pretende interrumpir un servicio del sistema, tanto a nivel físico como lógico. Un elemento del sistema es destruido o se hace inservible, por lo cual resulta una amenaza para la disponibilidad. Ejemplos: la destrucción de algún dispositivo de hardware (discos, líneas de comunicación, etc.) y los ataques DoS (Denegations of Service), aprovechando algunos bugs del protocolo o del sistema operativo.
- Intercepción: este ataque se produce cuando un intruso accede a un

elemento protegido y privado. Podemos citar como ejemplos el acceso a la información personal de los usuarios, a archivos y a documentos personales.

- **Modificación:** es un ataque que se identifica porque el intruso no sólo accede a información, sino que la modifica y adultera en forma maliciosa. Es un tipo de ataque que puede derivar en graves consecuencias, y es, fundamentalmente, una amenaza a la integridad. Ejemplos: alterar el contenido de un archivo (archivo de claves) o modificar un programa para que funcione de manera distinta de la normal (instalación de troyanos).
- **Creación:** se refiere a la inserción de nuevos elementos en el sistema por parte de una persona no autorizada. Es también una amenaza a la integridad. Ejemplos: adición de registros a un archivo e inclusión de mensajes espurios en una red.

Ataques de software

Se producen normalmente luego que un intruso tuvo acceso al sistema o cuando el accionar imprudente de usuarios o administradores implica exponerse a riesgos sin tomar los **recaudos** necesarios, por ejemplo cuando se ejecuta un software sin el previo chequeo y validación.

- **Bomba lógica:** es un código que se incrusta en un programa host o portador, que comprueba si ciertas condiciones se cumplen, en cuyo caso ejecuta alguna acción no autorizada, por lo general, con **consecuencias destructivas**. Estas condiciones pueden ser la existencia de ciertos ficheros, una fecha particular, la ejecución de una aplicación, etc. Una vez que la bomba explota, puede alterar o eliminar datos, parar el sistema, etc. Un ejemplo de uso de bomba lógica es el caso de un programador que vende un programa a una empresa. Si transcurrido un cierto tiempo la empresa no ha pagado, el programador revela la existencia de la bomba lógica con el fin de obtener su dinero.
- **Puerta falsa o trasera** (Trapdoor o Backdoor): es un punto de entrada secreto en un programa. Alguien que conoce su existencia puede obtener permisos de acceso sin tener que pasar por los mecanismos normales de autentificación. La puerta falsa es un código que **reconoce** alguna secuencia de entrada especial o se dispara si es ejecutado por cierto usuario o por la ocurrencia de una secuencia determinada.

nada de sucesos. Normalmente, en sistemas LINUX, estas puertas son demonios que se encuentran escuchando un determinado puerto a la espera de la señal correcta.

- **Caballo de Troya** (Trojan Horse): rutina oculta en un programa de utilidad. Cuando el programa se ejecuta, también lo hace la rutina, que realiza acciones no autorizadas y perniciosas. Estos programas permiten llevar a cabo, de forma indirecta, acciones que no se pueden realizar de forma directa. Por ejemplo, un Caballo de Troya puede ser un editor que, al ser ejecutado, modifica los permisos de los ficheros que edita, de manera que puedan ser accedidos por cualquier usuario.
- Virus: programa que se incrusta en otro, parasitándolo. Esta capacidad de reproducción ya es lo suficientemente nociva para el sistema, porque se genera código dañino o destructivo. Hay muy pocos virus en LINUX.
- Bacteria: programa que consume recursos del sistema replicándose a sí mismo, pero que no daña explícitamente ningún archivo. Se suele reproducir en forma exponencial, por lo que puede acaparar recursos de CPU, memoria y disco.
- Gusano (worm): es un virus que usa las redes de computadoras para pasar de un sistema a otro. Una vez que llega a un sistema, el gusano se puede comportar como un virus o como una bacteria, puede implantar programas del tipo Caballo de Troya o realizar acciones no autorizadas. Para replicarse, los gusanos emplean algunos programas que proporcionan servicios de red, como correo electrónico, ejecución remota de programas y conexión a sistemas remotos.

Valor intrínseco y derivado de las áreas protegidas

Discriminando las áreas de riesgo, podemos dar un valor ficticio a cada una usando una valoración de 0 a 100 para representar el costo que pueden ocasionar la pérdida, robo o adulteración de datos, como así también ataques de denegación de servicios.

Servidores de Internet

| | |
|------------------------------------|--|
| Tipo de ataque | |
| Valor | |
| Intrínseco | |
| Derivado | |
| Información de usuarios | |
| 30 | |
| 100 | |
| Interrupción del servicio por hora | |
| 50 | |
| 0 | |
| Robo de cuentas por unidad | |
| 20 | |
| 10 | |

Definición de política y estrategia

La política de seguridad que se propone está planteada fundamentalmente desde los puntos de vista de la vulnerabilidad digital y física. A tal efecto, abordaremos cada una de ellas por separado, considerando tanto los recursos necesarios en materia de hardware y software, como los humanos y físicos.

Procedimiento para la detección de intrusos

Éstos son los pasos que recomienda el CERT (www.cert.org):

- Examinar los ficheros log como el `last`, `log`, contabilidad, `syslog`, y los `C2 log` buscando conexiones inusuales o elementos sospechosos en el sistema. Hay que tener especial cuidado en guiarlos por los logs, ya que muchos intrusos utilizarán diversas herramientas para borrar sus huellas.
- Buscar por el sistema ficheros ocultos o inusuales (ficheros que empiezan por un `.` y no salen con un simple `ls`), ya que se los puede utilizar para esconder herramientas que romperán la seguridad del sistema (por ejemplo, un crackeador). También pueden contener el `/etc/passwd` del sistema o de otros sistemas a los cuales ha entrado

nuestro intruso. Muchos piratas suelen crear directorios ocultos utilizando nombres como **... (punto-punto-punto)**, **.. (punto-punto)** o **..^g (punto-punto control+G)**. En algunos casos, un pirata ha utilizado nombres como **.x**, **.hacker** o incluso **.mail**.

- Buscar ficheros SET-UID por el sistema, ya que, en muchas ocasiones, los piratas suelen copiar y dejar escondidas copias del **/bin/sh** para obtener root. Podemos utilizar el comando '**find**' para buscar este tipo de ficheros por el sistema (este comando puede ser sustituido por un troyano para esconder ficheros del pirata, por lo que no es totalmente fiable). Para ello, ejecutamos la siguiente línea:

```
# find / -user root -perm -4000 -print
```

- Revisar los ficheros binarios del sistema para comprobar que no han sido sustituidos por un troyano, como los programas **su**, **login**, **telnet** y otros programas críticos del sistema. (Existen varias herramientas conocidas como RootKit que permiten a un pirata cambiar los binarios del sistema por troyanos que son copias exactas de los originales.) Lo recomendado es comparar con las copias de seguridad, aunque puede que éstas también hayan sido sustituidas por un troyano.
- Examinar todos los ficheros que se ejecutan por cron y at, ya que algunos piratas depositan puertas traseras que les permiten volver al sistema aunque los hayamos echado. Tenemos que asegurarnos de que todos los ficheros son nuestros y no tienen permiso de escritura.
- Examinar el fichero **/etc/inetd.conf** en busca de cambios, en especial de aquellas entradas que ejecuten un shell (por ejemplo: **/bin/sh** o **/bin/csh**) y comprobar que todos los programas son legítimos del sistema y no troyanos.
- Examinar los ficheros del sistema y la configuración en busca de alteraciones. En particular, buscar entradas con el signo + o host names no apropiados en ficheros como **/etc/hosts.equiv**, **/etc/hosts.lpd** y en todos los ficheros **.rhost** del sistema, con especial interés los de root, uucp, ftp y otras cuentas. Estos ficheros no deberían tener atributo de escritura.
- Examinar cuidadosamente todos los ordenadores de nuestra red local en busca de indicios de que ha sido comprometida. En particu-

lar, revisar aquellos sistemas que comparten NIS+ o NFS, o los listados en el `/etc/hosts.equiv`. Lógicamente, también habrá que revisar los sistemas informáticos que los usuarios comparten mediante el acceso del `.rhost`.

- Examinar el fichero `/etc/passwd` en busca de alteraciones en las cuentas de los usuarios o la creación de cuentas nuevas, en especial, aquellas cuentas con ID 0, las que no tienen password, etc.

Servidores

Recursos

- **Hardware:** obviamente, mientras mejor sea la calidad de la computadora que usemos, más seguros estaremos de su infalibilidad. Claro que también representará mayor costo en caso de daño.
- **Software:** sólo hay que considerar la compra periódica de las instalaciones del sistema operativo LINUX de la distribución preferida; eventualmente, será necesario adquirir algún software propietario. El resto es despreciable como costo, ya que todo el software que se precisa es de libre distribución y se encuentra protegido por la licencia GPL/GNU.
- **Humanos:** en este punto, es importante destacar que no sólo hace falta un responsable en esta área, sino que también se necesita la participación de los niveles superiores para la toma de decisiones e instrumentación de las políticas.
- Físico: este punto deberá ser considerado de acuerdo con las cualidades del lugar donde se encuentre instalado el sistema. Habrá que tener en cuenta factores tales como accesibilidad de intrusos, cuestiones ambientales, etc.

Medidas Digitales

1. Diagnóstico del estado actual del sistema.
2. Instalación de un sistema de acceso remoto seguro (SSH).
3. Instalación de un sistema de control y consulta a través de la Web.
4. Uso de un sistema de encriptación para el correo de alto riesgo con sistema de llave pública (PGP) para los niveles superiores y de seguridad.

5. Actualización de las últimas versiones del sistema operativo y de los programas necesarios para mantener el servicio.
6. Monitoreo del estado y recursos del sistema.
 - Control de accesos.
 - Control de procesos.
 - Control de tráfico.
 - Logs de todos los accesos a los servidores.

Cronograma para estas actividades

| Actividad | Periodicidad |
|------------------|-------------------------------------|
| Diagnóstico | Mensual |
| Actualizaciones | Con la salida de cada nueva versión |
| Shell seguro | Permanente |
| Consulta por Web | Permanente |
| Encriptación | Permanente |
| Monitoreo | Permanente |

Físicas

- Se deberá establecer un perímetro de seguridad alrededor de las computadoras que cumplen la función de servidores.
- Habrá que considerar también los factores de riesgo ambiental y de catástrofes o accidentes tales como tormentas o incendios.
- Es preciso contar con un soporte seguro y adecuado de backup y recuperación de la información.
- Evitar el consumo de bebidas en proximidad de los equipos, como así también el acceso de niños.
- Sistema de alarmas y vigilancia.
- Extintores de incendios.

- Todos los papeles provenientes de la oficina gerencial y de las de los servidores deben ser destruidos antes de ser arrojados a los contenedores de basura.
- Ningún empleado ni personal brindará información a clientes por teléfono, teniendo especial cuidado en las claves.
- No se cambiarán claves de acceso a través de pedidos telefónicos.
- Ningún empleado, a excepción del encargado de seguridad y el webmaster, tendrá acceso a cuentas shell.
- No se proveerán cuentas shell a los usuarios que no las necesiten.
- No se deberán almacenar bajo ningún medio las claves de los usuarios sin encriptar.
- Se deberá establecer una política mensual de actualización de claves para los usuarios con cuentas shell.
- Un solo empleado será el responsable de la asignación de claves y sólo será reemplazado por el encargado de seguridad o uno de los gerentes.
- La selección de claves no se hará discrecionalmente ni a voluntad de los usuarios, sino que serán otorgadas por el personal designado usando un software desarrollado al efecto.

Medidas a tomar en caso de detección de intrusos

1. Inmediato aislamiento del área atacada.
2. Evaluación de daños.
3. Identificación del intruso.
4. Recopilación de pruebas.
5. De acuerdo con las características del intruso y el daño ocasionado, se deberá optar por una de las siguientes medidas:
 - Contacto personal con el intruso en caso de que no existan daños.
 - Contacto legal con el intruso en caso de corroborar la existencia de robo, daño o adulteración de información.
6. Información de la intromisión a los ISPs conocidos.
7. Denuncia ante los organismos correspondientes.
8. Evaluación de la necesidad de iniciar acciones legales.

EJERCICIOS

| | |
|---|------------------|
| Entrada en el sistema y consulta del sistema y de los usuarios | .Pág. 179 |
| Comunicaciones con otros usuarios | .Pág. 180 |
| Comandos básicos | .Pág. 181 |

A

Apéndice

Entrada en el sistema y consulta del sistema y de los usuarios

Objetivos:

- Que el usuario se familiarice con las rutinas de entrada y salida del sistema.
- Que el usuario sea capaz de solicitar e interpretar información básica del sistema.

Ejercicio 1

Inicie LINUX y regístrese como root.

Ejercicio 2

Consulte información personal usando los comandos:

who am i
logname
id

Ejercicio 3

Consulte información sobre el sistema operativo con los comandos:

uname
uname -v
uname -a

Ejercicio 4

Cambie su clave de root con el comando **passwd**, luego salga del sistema y vuelva a loguearse usando su nueva clave.

Ejercicio 5

Cree un nuevo usuario llamado pepe con el comando **adduser**.

Ejercicio 6

Asígnele una clave al usuario pepe con el comando **passwd**.

Ejercicio 7

Inicie otra sesión en una terminal virtual y loguéese como pepe.

Ejercicio 8

Consulte qué usuarios se encuentran conectados al sistema con el comando **who**.

Ejercicio 9

Consulte la fecha y la hora del sistema con el comando **date**.

Ejercicio 10

Consulte la fecha con los siguientes parámetros y analice la salida:
date '+Son las %T del da %e %n del mes %m del anio %y'

Ejercicio 11

Use el comando **cal** para visualizar el calendario del año 1999.

Ejercicio 12

Consulte el qué día de la semana cae el 01 de noviembre del año 2000.

Ejercicio 13

Use el comando **finger** para consultar los datos del usuario root.

Comunicaciones con otros usuarios

Objetivos:

- Que el usuario pueda comunicarse con los otros usuarios mediante las herramientas básicas del sistema.

Ejercicio 1

Logueado como root, use el comando **mail** para consultar su correo.

Ejercicio 2

Use las teclas * y - para leer el mensaje anterior y el posterior al actual, respectivamente.

Ejercicio 3

Envíe un mail a pepe con el comando **mail pepe@hostname**.

Ejercicio 4

Envíe correo a un usuario inexistente y observe qué sucede.

Ejercicio 5

Loguéese en otra terminal como pepe.

Desde la terminal donde está como root, use el comando **write** para enviarle un mensaje a pepe.

Desde la terminal donde está como pepe, observe el mensaje.

Desactive la recepción de mensaje con el comando **mesg n** del usuario pepe.

Desde la terminal donde está como root, nuevamente use el comando **write** para enviarle un mensaje a pepe. Observe lo que sucede.

Active la recepción de mensaje con el comando **mesg y** del usuario pepe.

Comandos básicos

Ejercicio 1

Mediante el comando **echo**, visualice en pantalla la siguiente frase: "ésta es una frase de prueba".

Ejercicio 2

Visualice la misma frase pero en dos líneas usando el carácter de control \n.

Ejercicio 3

Ingresé las siguientes líneas:

```
echo "Mi grupo es\c";id -g -n;echo "que se corresponde con  
el numero\c";id -g.
```

```
echo "Mi directorio actual es: \c";pwd
```

```
echo "Listado de las personas conectadas: "; finger  
banner "Mi id es `logname`"
```

Observe y analice la salida.

Ejercicio 4

Use los comandos **basename \$PWD** y **dirname \$PWD** para ver info sobre los directorios.

Ejercicio 5

Explique en pocas palabras cuál es la diferencia entre la referencia absoluta y la relativa a un directorio dado.

Ejercicio 6

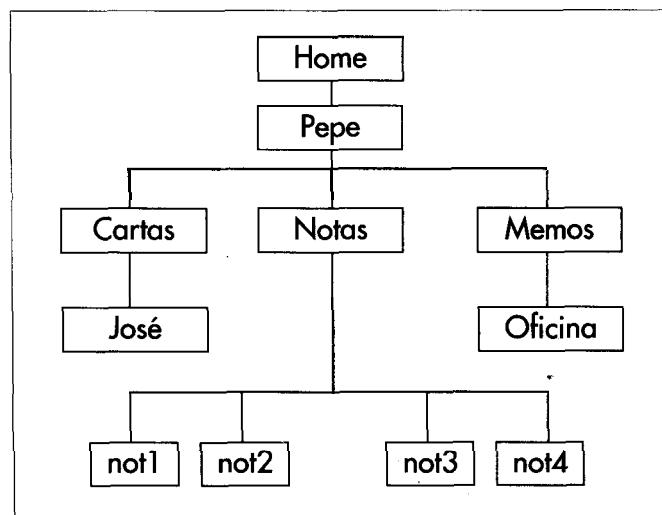
¿En dónde se crea el directorio de cada usuario nuevo del sistema?

Ejercicio 7

Solicite el listado de su directorio home usando la variable \$HOME y ~.

Ejercicio 8

Cree la siguiente estructura de directorios:

**Ejercicio 9**

Desde el directorio **not4**, en la estructura antes creada, haga las siguientes consultas de directorios:

- Contenido del directorio **not2** en forma absoluta.
- Contenido del directorio **oficina** en forma relativa.
- Listado recursivo de directorios desde **/home** usando la opción -R.
- Cambie al directorio **jose** desde **not4**.
- Use los comandos **cd \$HOME cd -** y observe los resultados.
- Use el comando **ls -l** y describa la información suministrada.

Ejercicio 10

Use los comandos **cat**, **less** y **more** para visualizar el contenido del archivo **profile** en el directorio **/etc**.

Ejercicio 11

Copie todos los archivos del directorio **/etc/ppp** al directorio **/home/pepe**.

Ejercicio 12

Mueva los archivos del directorio **/home/pepe** al directorio **/home/pepe/cartas/jose**.

Ejercicio 13

Borre los archivos de **/home/pepe/cartas/jose** uno por uno.

Ejercicio 14

Borre el directorio **notas** de **/home/pepe** con todo su contenido.

Ejercicio 15

Cree un enlace simbólico en **/home/pepe** llamado nuevo a **/home/pepe/memos**.

LICENCIA GENERAL PÚBLICA GNU

| | |
|-----------------|----------|
| Preámbulo | Pág. 187 |
| Términos | Pág. 188 |

Apéndice

B

Licencia general pública GNU

Versión 2, junio de 1991 Copyright@ 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139 USA

Se permite a todo el mundo copiar y distribuir copias idénticas de este documento de licencia, aunque no se permite su modificación.

Preámbulo

Las licencias de uso de casi todo el software han sido establecidas para quitarle la libertad de compartirlo y modificarlo. En contraste con esta costumbre, la Licencia General Pública GNU pretende garantizar su libertad de compartir y modificar software gratuito a fin de asegurar que éste sea de uso libre para todos sus usuarios. Esta Licencia General Pública es de aplicación para la mayor parte del software de la *Free Software Foundation* y para cualquier otro programa cuyos autores se comprometen a su utilización. (Otro software de la *Free Software Foundation* está cubierto alternativamente por la Licencia General Pública de biblioteca GNU.) También puede aplicar ésta a sus programas.

Cuando hablamos de software gratuito, nos referimos a la libertad, no al precio. Nuestras Licencias Generales Públicas han sido diseñadas para asegurarle la libertad necesaria para distribuir copias de software gratuito (y de cobrar por este servicio, si así lo desea), para que reciba código fuente o pueda obtenerlo si lo desea, para que pueda modificar el software o utilizar partes del mismo en nuevos programas gratuitos y para que sepa que puede hacer estas cosas.

A fin de proteger sus derechos, es necesario constituir restricciones que impidan que alguien pueda denegarle estos derechos o pedirle que renuncie a los mismos. Estas restricciones implican también ciertas responsabilidades si distribuye copias del software o lo modifica. Por ejemplo, si distribuye copias de este programa, tanto gratis o con ánimo de lucro, deberá conceder a los recipientes del mismo todos los derechos con los que cuente. Deberá asegurarse de que los recipientes reciben o pueden obtener el código fuente. Y deberá asimismo mostrarles estos términos, a fin de que puedan conocer sus derechos. Protegemos sus derechos con dos pasos: (1) copyright del software y (2) le ofrecemos esta licencia que le concede permiso legal para copiar, distribuir y/o modificar el software.

Además, para la protección de cada autor así como para la nuestra

propia, deseamos asegurarnos de que todo el mundo comprende que este software gratuito no cuenta con garantía alguna. Si el software es modificado por alguna otra persona y posteriormente distribuido deseamos que sus destinatarios sepan que lo que reciben no es el producto original, de forma que cualquier problema introducido por otras partes no se refleje en la reputación profesional de los autores originales. Finalmente, cualquier programa gratuito se ve constantemente amenazado por patentes de software. Deseamos evitar el peligro de que los redistribuidores de un programa gratuito obtengan licencias de patente a título individual, lo que haría de éste un programa privado. A fin de impedir esto, hemos dejado muy claro que cualquier patente obtenida deberá ser licenciada para la utilización gratuita de todos o no deberá ser licenciada en absoluto.

TÉRMINOS

TÉRMINOS Y CONDICIONES DE LA LICENCIA GENERAL PÚBLICA DE GNU PARA LA REALIZACIÓN DE COPIAS, DISTRIBUCIÓN Y MODIFICACIÓN

1. Esta Licencia es de aplicación a cualquier programa u otro trabajo que contenga un aviso incluido por el titular de los derechos de copyright estableciendo su libertad de distribución bajo los términos de esta Licencia General Pública. El "Programa", a continuación, hace referencia a cualquier programa o trabajo de este tipo y un "trabajo basado en el Programa" indica el Programa o cualquier tipo de trabajo derivado bajo las leyes de copyright: es decir, un trabajo que contendría el Programa o una porción del mismo, bien en versión exacta o bien con modificaciones y/o traducido a otro idioma. (En adelante, las traducciones quedan incluidas, sin limitación alguna, en el término "modificación".) Cada beneficiario de esta licencia quedará indicado por el término "usted". Las actividades que no incluyan copia, distribución y modificación no quedarán protegidas por esta Licencia; quedan fuera de su alcance. El acto de ejecución del Programa no queda limitado y los resultados obtenidos de la ejecución del programa quedan cubiertos sólo si su contenido constituye un trabajo basado en el Programa (independientemente de que haya sido obtenido mediante la ejecución del Programa). La validez de esto dependerá de lo que haga el Programa.

2. Podrá copiar y distribuir copias exactas del código fuente del Programa según lo hubiera recibido, utilizando cualquier medio, siempre que, de forma notoria y apropiada, publique, en cada copia, un aviso adecuado de derechos de copyright relativo a la ausencia de garantía alguna y de que proporcione a todos los destinatarios del Programa una copia de esta Licencia junto con el mismo. Podrá cobrar honorarios por el acto físico de transferencia de una copia y podrá, a su propia discreción, ofrecer servicios de protección de garantía a cambio de honorarios.
3. Puede modificar su copia o copias del Programa o cualquier porción del mismo, hasta la formación de un trabajo basado en el Programa, y podrá asimismo copiar y distribuir dichas modificaciones o trabajos bajo los términos de la Sección 1 anteriormente reseñada, siempre que cumpla las condiciones adicionales que se detallan a continuación:
 - a) Deberá asegurarse de que los archivos modificados incorporen avisos prominentes que establezcan esta modificación, así como la fecha en que fue realizada.
 - b) Deberá asegurarse de que cualquier trabajo que distribuya o publique, que pudiera, en su totalidad o en parte, contener o haber sido derivado del Programa o de cualquier parte del mismo, sea licenciado en su totalidad y sin cargo alguno a terceros que lo requieran bajo los términos de esta Licencia.
 - c) Si el programa modificado lee normalmente comandos interactivamente cuando se ejecuta, deberá asegurarse de que, cuando se ponga en marcha su ejecución para dicha utilización interactiva de la forma mas común, imprima o muestre una proclama que incluya un aviso correspondiente relativo a los derechos de copyright y una clarificación de la falta de garantía (o estableciendo la provisión de dicha garantía por su parte) y que especifique que los usuarios pueden redistribuir el programa bajo estas condiciones, e informe al usuario cómo visualizar una copia de esta Licencia. (Excepción: si el Programa es de por sí interactivo pero no imprime normalmente dicha proclama, tampoco se requerirá que su trabajo basado en el Programa deba imprimirla.) Estos requisitos son de aplicación al trabajo modificado en su totalidad. Si hubiera secciones identificadas de dicho trabajo no derivadas del Programa y que pudieran considerarse trabajos razonablemente independientes, entonces esta Licencia, así como sus términos, no serían de aplicación a dichas secciones cuando las distribuya como trabajos independientes. Pero cuando

distribuya las mismas secciones como parte de un trabajo completo mayor basado en el Programa, la distribución del trabajo completo deberá realizarse bajo los términos de esta Licencia, cuyos permisos con respecto a otras licencias se ampliarán a la totalidad del trabajo y, por lo tanto, a cada una de sus partes y respectivamente a quien lo ha escrito. Por todo esto, no es el propósito de esta sección reclamar derechos o disputar sus derechos a trabajos escritos enteramente por el titular de esta licencia, sino ejercitar el derecho a controlar la distribución de trabajos derivados o colectivos basados en el Programa. Además, la simple toreado(Sn de otro trabajo no basado en el Programa a éste (o con no trabajo basado en el Programa) en un volumen de un medio de almacenamiento o distribución no somete al otro trabajo al alcance de esta Licencia.

4. Podrá copiar y distribuir el Programa (o un trabajo basado en el mismo, bajo los términos de la Sección 2) en código objeto o formato ejecutable bajo los términos anteriormente reseñados en las Secciones 1 y 2, siempre que observe asimismo una de las condiciones siguientes:
 - a) Lo acompañe con el correspondiente código completo legible por el sistema, que deberá distribuirse bajos los términos anteriormente reseñados en las Secciones 1 y 2 en un medio normalmente utilizado para intercambios de software; o
 - b) Lo acompañe con una oferta por escrito, con una validez mínima de tres años, de facilitar a terceros, a un precio no superior al costo de realizar la distribución física del código, una copia completa legible por la máquina del correspondiente código fuente, a distribuir bajo los términos antes reseñados en las Secciones 1 y 2 en un medio normalmente utilizado para intercambios de software; o
 - c) Lo acompañe con la información que haya recibido en relación a la oferta de distribución del código fuente. (Esta alternativa se permite exclusivamente para distribuciones no comerciales y sólo si ha recibido el programa en código objeto o formato ejecutable con dicha oferta, de acuerdo con los términos anteriormente reseñados en la Sub-sección b.) El código fuente de un trabajo significa la forma preferida del trabajo para la realización de modificaciones al mismo. Para un trabajo ejecutable, el código fuente completo significa todo el código fuente para todos los módulos que contiene, más cualquier archivo de definiciones de interfase asociados, más las secuencias que se utilicen para controlar la compilación y ejecución del ejecu-

table. Sin embargo, como una excepción especial, el código fuente distribuido no necesita incluir nada que sea normalmente distribuido (en formato fuente o binario) con los componentes principales (compilador, kernel, etc.) del sistema operativo en el que se ejecuta el ejecutable, a menos que el propio componente acompañe al ejecutable. Si llegara a realizarse una distribución de código objeto o ejecutable ofreciendo acceso a copiar de un lugar designado, el ofrecimiento de acceso equivalente a copiar el código fuente del mismo lugar contará como una distribución del código fuente, incluso cuando las terceras partes no tengan que copiar el código fuente junto con el código objeto.

5. No podrá copiar, modificar, sublicenciar o distribuir el Programa salvo en la forma en que esta Licencia expresamente lo permita. Cualquier otro intento de copia, modificación, sublicencia o distribución del Programa será nula y dará fin de forma automática a sus derechos bajo los términos de esta Licencia. Sin embargo, las licencias correspondientes a terceros que hubieran podido recibir copias o derechos del titular de esta licencia bajo los términos de la misma no quedarán canceladas siempre que dichas partes **observen** sus condiciones.
6. No tiene obligación alguna de aceptar esta Licencia, ya que no ha firmado la misma. Sin embargo, nada aquí le concede permiso de modificación o distribución del Programa o de sus trabajos derivados. Estas acciones quedarán prohibidas por la ley si decidiera no aceptar esta Licencia. Por lo tanto, mediante la modificación o distribución del Programa (o de cualquier trabajo basado en el mismo), indicará esta Licencia y de sus términos y condiciones para la copia, distribución o modificación del Programa o de los trabajos basados en el mismo.
7. Cada vez que redistribuya el Programa (o cualquier trabajo basado en el mismo), el destinatario recibe automáticamente una licencia del titular original para copiar, distribuir o modificar el Programa en base a estos términos y condiciones. No podrá imponer restricción adicional alguna sobre el ejercicio, por parte del destinatario, de los derechos que aquí se conceden. No tendrá responsabilidad alguna en lo que atañe a asegurar la observación, por terceros, de las condiciones a las que está sujeta esta licencia.
8. Si, como consecuencia de una sentencia judicial de infracción de derechos de patente o por cualquier otra razón (no limitada a cuestiones de patente), se le impusiera condición alguna (por sentencia ju-

dicial, acuerdo, o cualquier otra vía) que estuviera en contradicción con las condiciones de esta Licencia, dicha condición no podrá constituir excusa en lo que atañe a la obligación de observar las condiciones de esta Licencia. Si no pudiera efectuar distribuciones de forma que satisfaga simultáneamente sus obligaciones bajo los términos de esta Licencia y cualesquiera otras obligaciones que le pudieran corresponder, la consecuencia de dichas circunstancias es que no podrá distribuir el Programa en absoluto. Por ejemplo, si una licencia de patente no permitiera la redistribución libre de derechos de royalty del Programa por parte de aquellos que recibieran copias de forma directa o indirecta a través suya, entonces la única forma en que podría satisfacer dicha licencia de patente y esta Licencia consistiría en no distribuir el Programa en absoluto. Si cualquier porción de esta sección fuera declarada inválida o no ejecutable bajo cualquier circunstancia determinada, el resto de la sección seguiría siendo de aplicación y la sección completa seguiría siendo asimismo aplicable siempre que concurran otras circunstancias. No es el propósito de esta sección inducirle a infringir patentes u otros derechos de la propiedad intelectual o disputar en forma alguna la validez de dichos derechos; el único objetivo de esta sección es la protección de la integridad del sistema de distribución de software gratuito, implantado por prácticas de licencia pública. Muchas personas han hecho generosas contribuciones a una amplia gama de software distribuido a través de este sistema confiando en la coherente aplicación de este sistema de distribución; depende del autor/donante decidir si desea distribuir el software mediante cualquier otro sistema, y un licenciado no puede establecer imposición alguna con respecto a dicha decisión. Esta sección pretende aclarar perfectamente lo que se supone es una consecuencia del resto de esta licencia.

9. Si la distribución y/o utilización del Programa quedara restringirla a ciertos países bajo el efecto de derechos de patente o interfaces a copyright, el titular original de los derechos de copyright que ampara el Programa bajo esta Licencia podrá añadir una limitación explícita del ámbito de distribución geográfica que excluya a dichos países, de forma que la distribución quede permitida sólo en países, o entre países, que no estén sujetos a dicha exclusión. En este caso, esta Licencia incorporará la limitación de la misma forma que si hubiera sido incluida en el texto de esta Licencia.
10. La Free Software Foundation podrá publicar versiones revisadas y/o nuevas de la Licencia General Pública de forma periódica. Estas nue-

vas versiones serán similares en espíritu a la versión actual, aunque podrían diferir en detalles a fin de resolver nuevos problemas o cuestiones. Cada versión recibirá un número distinto. Si el Programa especifica un número de versión de esta Licencia que sea de aplicación a la misma y a "cualquier otra versión posterior", tendrá la opción de seguir los términos y condiciones de dicha versión o de cualquier versión posterior publicada por la Free Software Foundation. Si el Programa no especificara un número de versión de esta Licencia, podrá seleccionar cualquier versión publicada en cualquier momento por la Free Software Foundation.

11. Si deseara incorporar partes del Programa en otros programas gratuitos cuyas condiciones de distribución sean distintas, escriba al autor y solicite permiso. Escriba a la Free Software Foundation con respecto al software cuyos derechos de copyright correspondan a esta organización; algunas veces concedemos excepciones a estas normas. Nuestra decisión será guiada por los dos objetivos de conservar el carácter gratuito de todos los derivados de nuestro software gratuito y de promover generalmente la participación en el software y su reutilización.
12. DEBIDO A QUE EL PROGRAMA SE LICENCIA LIBRE DE COSTO ALGUNO, EN LO QUE PERMITE LA LEY, ESTE PROGRAMA NO ESTÁ PROTEGIDO POR GARANTÍA ALGUNA. SALVO CUANDO SE ESTABLEZCA LO CONTRARIO POR ESCRITO, LOS TITULARES DE LOS DERECHOS DE COPYRIGHT Y/U OTRAS PARTES SUMINISTRAN EL PROGRAMA 'TAL Y COMO ES', SIN GARANTÍA DE NINGÚN TIPO, EXPRESA O IMPLÍCITA. INCLUYENDO PERO SIN QUE ESTO SIRVA DE LIMITACIÓN, LAS GARANTÍAS IMPLÍCITAS DE COMERCIABILIDAD O UTILIDAD PARA ALGÚN PROPOSITO DETERMINADO, LA TOTALIDAD DEL RIESGO EN RELACIÓN A LA CALIDAD Y RENDIMIENTO DEL PROGRAMA DESCANSAN EN EL USUARIO. SI SE LLEGARA A ENCONTRAR QUE EL PROGRAMA ADOLECE DE ALGÚN DEFECTO, EL USUARIO DEBERÁ ASUMIR TODOS LOS COSTOS QUE RESULTARAN NECESARIOS PARA SU SERVICIO, REPARACIÓN O CORRECCIÓN.
13. BAJO NINGUNA CIRCUNSTANCIA, A MENOS QUE LO REQUIERA ALGUNA LEY APPLICABLE O QUE SEA EXPRESADO POR ESCRITO, EL TITULAR DE LOS DERECHOS DE COPYRIGHT O CUALQUIER OTRA PARTE QUE HUBIERA PODIDO MODIFICAR Y/O REDISTRIBUIR EL PROGRAMA DE ACUERDO CON LAS CONDICIONES ANTERIORMENTE RESEÑADAS, SERÁ RESPON-

SABLE ANTE EL USUARIO POR DAÑOS Y PERJUICIOS, INCLUYENDO DAÑOS Y PERJUICIOS GENERALES, ESPECIALES, INCIDENTALES O CONSECUENTES, RESULTANTES DE LA UTILIZACIÓN O INCAPACIDAD DE USO DEL PROGRAMA (INCLUYENDO, PERO SIN QUE ESTO CONSTITUYA UNA LIMITACIÓN, LA PÉRDIDA DE DATOS O LA INEXACTITUD DE DICHOS DATOS O LAS PÉRDIDAS SOSTENIDAS POR EL USUARIO O TERCEROS O UN FALLO DEL PROGRAMA CON RESPECTO A SU FUNCIONAMIENTO CON OTROS PROGRAMAS), INCLUSO SI DICHO TITULAR U OTRA PARTE HUBIERA SIDO INFORMADO CON RESPECTO A LA POSIBILIDAD DE DICHOS DAÑOS Y PERJUICIOS.

FIN DE LOS TÉRMINOS Y CONDICIONES

Cómo aplicar estos términos a sus nuevos programas:

Si llegara a desarrollar un nuevo programa y deseara que éste fuera de la mayor utilidad posible al público, la mejor forma de conseguirlo es escribir software gratuito que todo el mundo pueda redistribuir y cambiar libremente bajo estos términos.

Para hacer esto, adjunte los avisos que se relacionan a continuación al programa. Lo más seguro es adjuntarlos al principio de cada archivo fuente a fin de expresar con la mayor efectividad la exclusión de garantías; y cada archivo debería tener por lo menos la línea de "copyright" y un indicador que informe sobre dónde puede localizarse el texto completo.

Copyright (C) 19yy <name Of author> rama es software gratuito; lo puede redistribuir y/o modificar bajo los términos de la Licencia General Pública GNU según lo publicado por la Free Software Foundation, bien la versión 2 de la Licencia o (a su propia opción) cualquier otra versión posterior.

Este programa se distribuye con la esperanza de que será útil, pero **SIN NINGÚN TIPO DE GARANTÍA**; incluso sin la garantía implícita de **COMERCIABILIDAD** o **UTILIDAD PARA ALGÚN PROPÓSITO DETERMINADO**. Véase la Licencia Pública General de GNU para más información.

Debería haber recibido una copia de la Licencia General Pública GNU junto con este programa; en caso contrario, escriba a la Free Software Foundation, Inc., 675 Mass Ave, Cambridge MA 02139

EE.UU. Adjunte también información sobre cómo contactar¹⁰ por e-mail o correo normal.

Si el programa es interactivo, haga que proporcione un corto aviso como el que se muestra a continuación cuando comience en modalidad interjectiva:

Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This free software, and you are welcome to redistribute it under certain conditions 1 type 'show c' for details.

Los comandos hipotéticos 'show w' y 'show c' deberían mostrar las secciones correspondientes de la Licencia General Pública. Como es natural, los comandos que utilice podrían denominar algo diferente a 'show w' y 'show c', podrían incluso ser pulsaciones de los botones del mouse o elementos de menú, lo que se adecue más a su programa.

También debería hacer, si lo creyera necesario, que su empresa (si trabaja como un programador) o su institución educativa, si correspondiera, firmara un "aviso de no aplicabilidad de derechos de copyright". A continuación se muestra un ejemplo; modifique los nombres de acuerdo con sus requisitos:

Yoyodyne, Inc. , hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker. <signature of Ty Coon> 1 April 1989 Ty Coon, President of Vice

Esta Licencia General Pública no permite la incorporación de su programa a programas privados. Si su programa es una biblioteca de subrutinas, podría considerar más útil permitir el enlace de aplicaciones privadas con la biblioteca. Si esto es lo que desea hacer, utilice la Licencia General Pública de Biblioteca GNU en lugar de esta Licencia.

DICCIONARIO DE PALABRAS CASTELLANIZADAS

Apéndice



Diccionario de **alabré** **ellanizad**

@: at, arroba, indica pertenencia.

address: dirección.

assembler: ensamblador.

attach: hacer un attachment, atachear, adjuntar, unir.

automatic shutdown: shutdown automático, cierre automático.

backup: hacer un backup, backupear, hacer un archivo de reserva.

background program: programa background, programa subordinado.

background: fondo.

backspace: retroceder.

batch job: trabajo batch, trabajo en serie, trabajo por lotes.

boot: butear, iniciar.

browse: browsar, hojear, curiosear.

buffer: registro intermedio.

bug: bicho, error de programación o compilación.

byte: octeto.

chat: chatear, conversar, charlar (vía Internet, relay chat).

clear: hacer un clear, borrar, despejar, limpiar la pantalla.

click: clic, clipear, chasquear.

clone: clon.

command: comando, mandato, orden.

compile: compilar.

compiler: compilador.

configuration: configuración.

control panel: panel de control.

debug: hacer un debug, depurar, limpiar, corregir errores.

delete: deletear, borrar, suprimir.

directory: directorio.

disk, disc: disco.

disk driver: disquetera.

diskette: disco flexible.

display: visual, pantalla.

domain: dominio.

download: hacer un download, downloadear, descender, bajar archivos.

drag: dragear, arrastrar.

e-mail: correo electrónico.

e-mail address: dirección de correo electrónico.

exit: salida, salir.

fíe: archivo, fichero.

floppy disk: disco flexible.

font: fuente, símbolo de caligrafía.

foreground: primer plano, por encima.

format: formatear, dar formato.

ftp: Protocolo de Transferencia de Archivos.

gate: puerta.

gateway: puerta, pórtico.

hacker: computoadicto, experto en burlar seguridad.

halt: parada.

hard disk: disco rígido.

hardware: equipo físico.

header: cabecera.

home page: página de inicio.

icon: ícono.

initialize: inicializar.

input: ingresar.

interface: interfase.

internet: la Red.

job: trabajo.

keyboard: teclado.

keyword: palabra clave.

load: cargar.

log in: loguear, hacer un log-in, comenzar el modo de diálogo.

log off: desloguear, hacer un log-off, terminar el modo de diálogo.

log on: comenzar la sesión.

log out: terminar la sesión.

log: logear, registrar.

loop: bucle.

monitor: monitor, consola.

mount the fíesystem: montar el sistema de archivos.

output: hacer un output, salir.

overflow: desbordamiento.

password: clave secreta, contraseña.

port: puerto, salida.

prompt: guía, símbolo de la línea de comandos.

quit: salir.

reboot: rebootear, reanudar, hacer un rebut, rebutear, reiniciar.

release: libertar, soltar.

reset: hacer un reset, resetear, reestablecer, reajustar.

restart: hacer un restart, reanudar.

route: ruta.

scan: escanear, explorar.

screen: pantalla.

server: servidor.

setup: alistar, preparar.

shutdown: hacer un shutdown, cierre.

spool: spoolear, bobinar.

startup: inicializador.

switch: conmutador.

task: tarea.

telnet: telnetear, comunicarse vía telnet.

upgrade: hacer un upgrade, subir de grado, actualizar.

upload: hacer un upload, uploadear, ascenso de archivos, subir.

window: ventana.

workstation: estación de trabajo.

LINUX EN INTERNET

| | |
|----------------------|----------|
| Distribuciones | Pág. 205 |
| Software | Pág. 210 |
| Noticias | Pág. 213 |
| Varios | Pág. 217 |

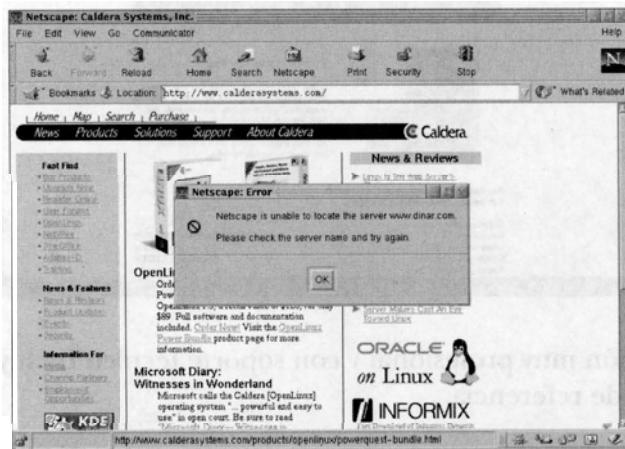
Apéndice

D

Linux en Internet

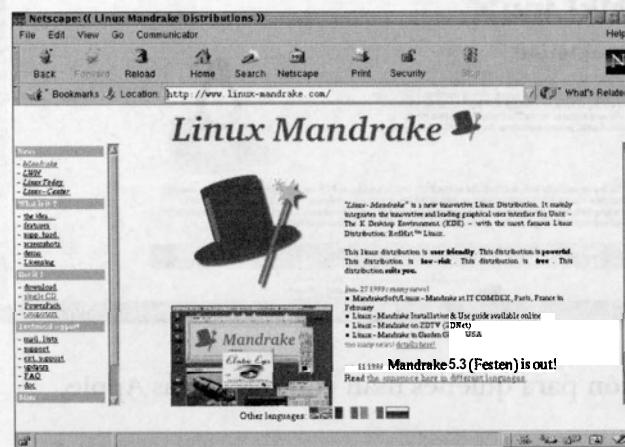
Distribuciones

CALDERA: www.calderasystems.com/products/openlinux/powerquest-bundle.html



Distribución comercial orientada a empresas y grandes usuarios; es una buena opción y tiene paquetes de software adicionales.

MANDRAKE: www.linux-mandrake.com/



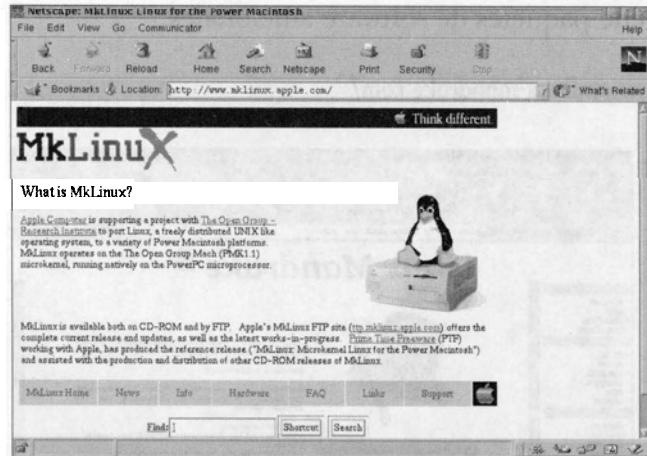
Distribución basada en Red Hat. Es una de las más actualizadas y resulta totalmente compatible con Red Hat en el sistema de paquetes RPM.

LINUX PRO PLUS: www.LinuxMall.com/Allprod/lxcdr.html



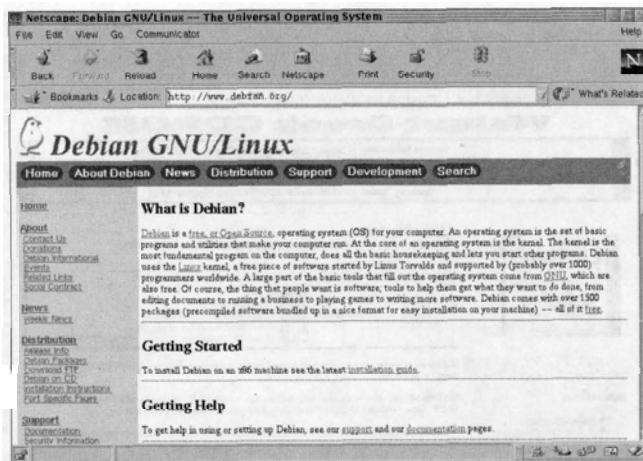
Una distribución muy profesional y con soporte técnico Incluye 6 CDs y una completa guía de referencia.

MKLINUX: www.mklinux.apple.com/



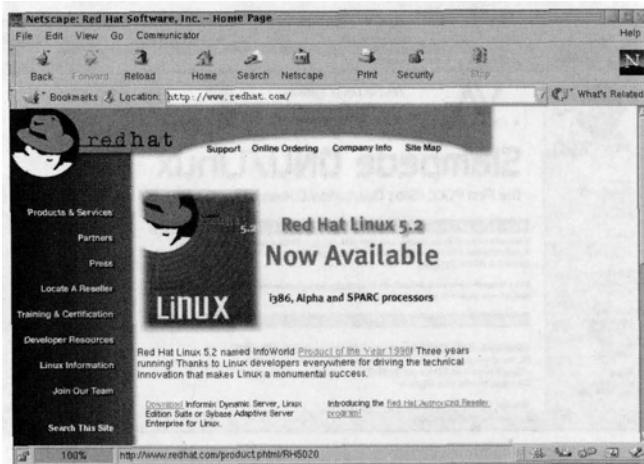
Una distribución para quienes usan computadoras Apple.

DEBIAN: www.debian.org/



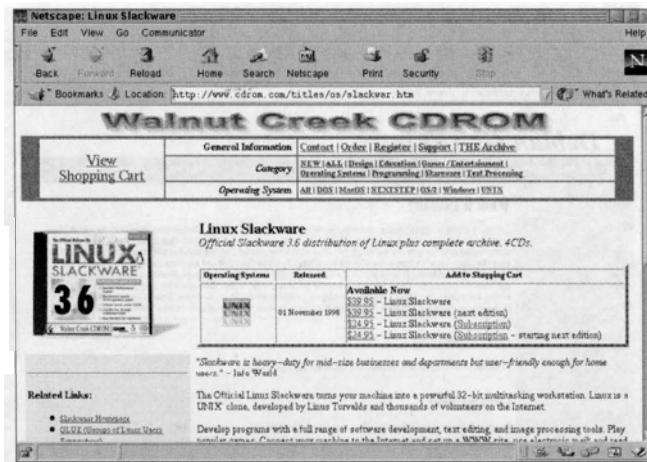
Una distribución completamente gratuita, muy completa y estable. Viene con más de 1500 paquetes (software precompilado y en un formato apto para instalar en su máquina) con formato propio.

RED HAT: www.redhat.com



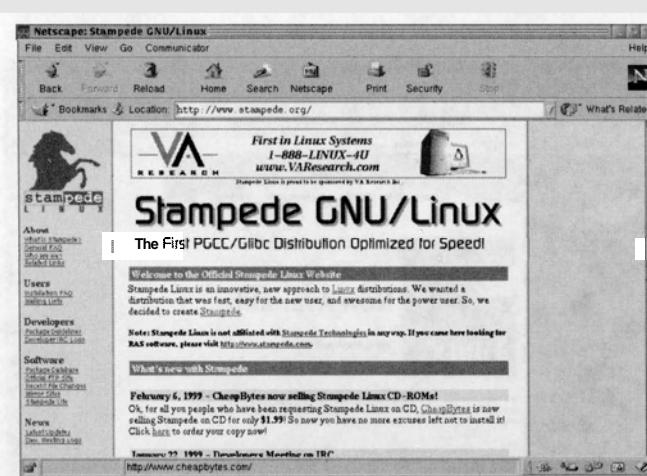
La más popular y usada de las distribuciones, por su facilidad de instalación, estabilidad y la gran biblioteca de programas disponibles en su sistema de paquetes (RPM).

SLACKWARE: www.cdrom.com/titles/os/slackwar.htm



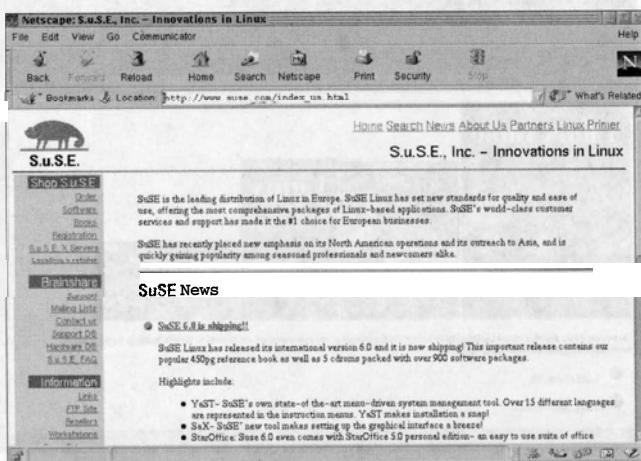
Una de las distribuciones más antiguas, muy estable y completa. Muchos veteranos Linuxeros se iniciaron con ella.

STAMPEDE: <http://www.stampede.org/>



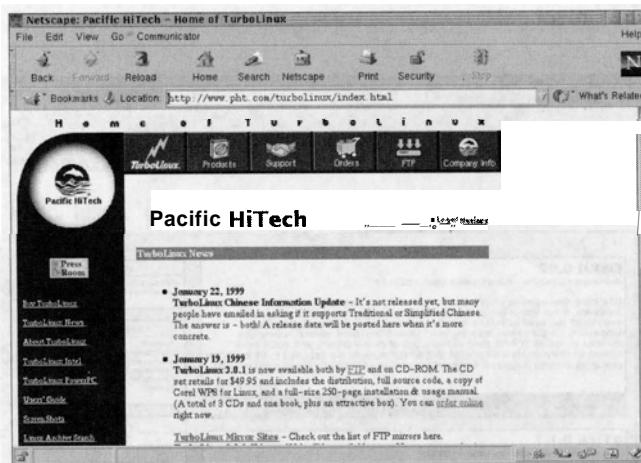
Una nueva distribución que pretende facilitar el uso y la instalación para usuarios nuevos.

S.u.S.E.: www.suse.com/index_us.html



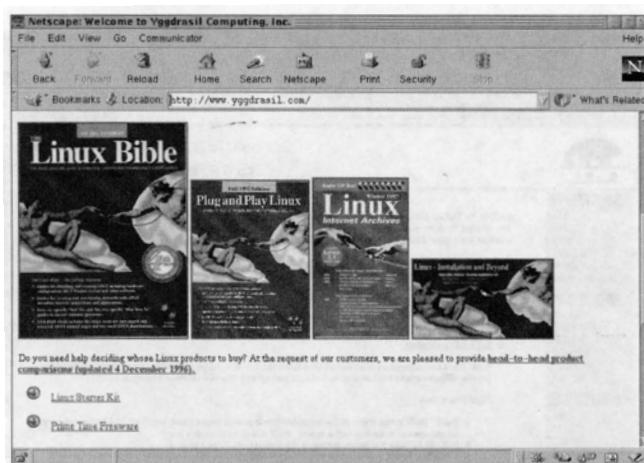
Distribución alemana (también está la versión en inglés) que Últimamente ganó muchos adeptos. Es una de las grandes colaboradoras de la causa GNU.

TURBO LINUX: www.pht.com/turboLinux/index.html



Es una distribución que contiene la base del sistema operativo más una colección de aplicaciones y documentación; está disponible para plataforma iX86.

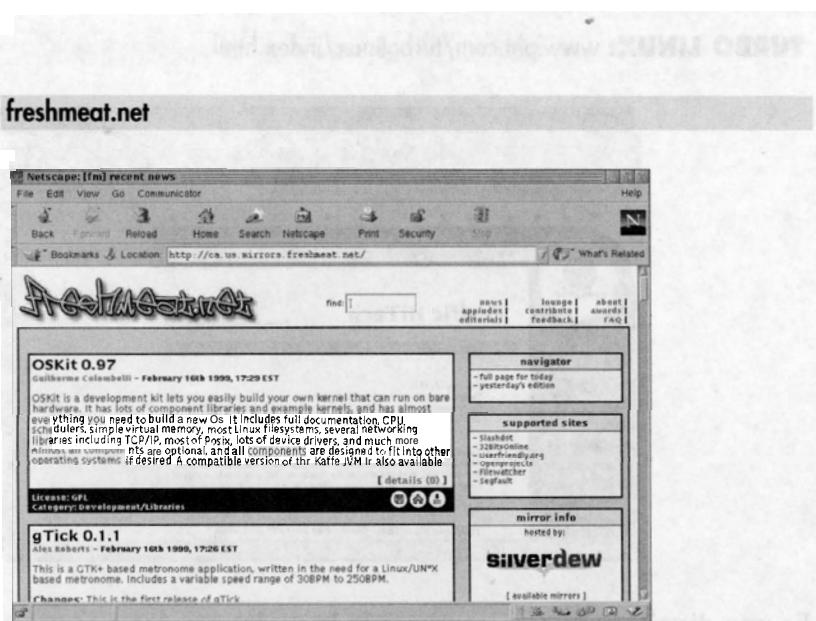
YGGDRASIL: www.yggdrasil.com/



La más veterana de las distribuciones. A ella se le debe gran parte de la difusión de LINUX

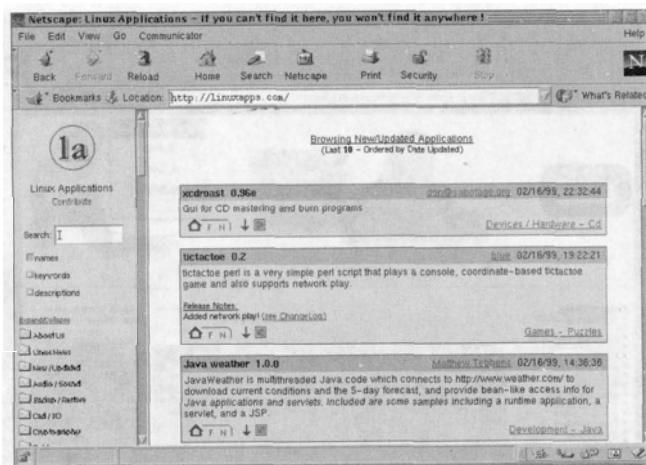
Software

FRESHMEAT: [freshmeat.net](http://ca.us.mirrors.freshmeat.net/)



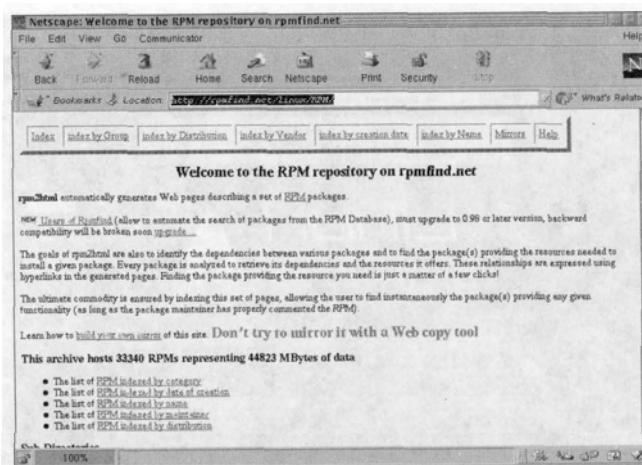
Si quieren mantenerse actualizados en cuanto a las últimas novedades en software para Linux, éste es el lugar por excelencia. Se actualiza día a día.

LINUX APPLICATIONS: linuxapps.com/



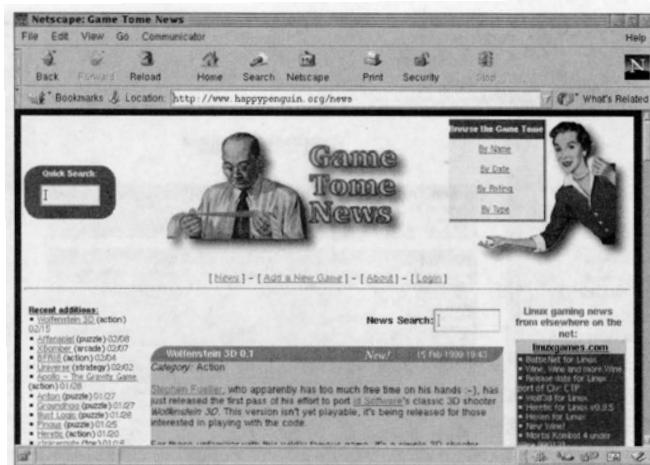
Un sitio con abundante material, muy bien catalogado y actualizado.

RPM REPOSITORY: rpmfind.net/linux/RPM/



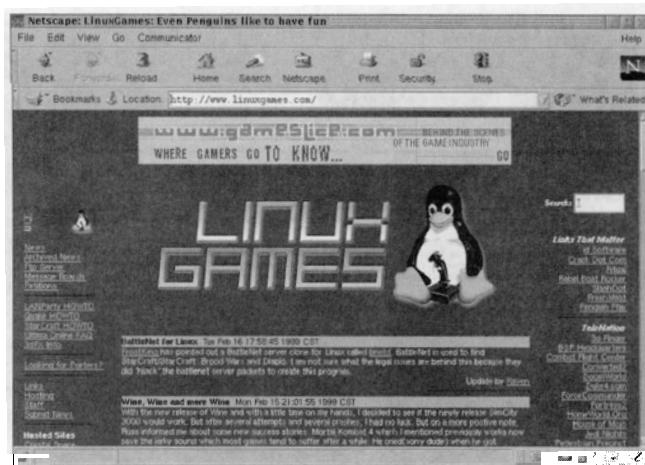
El más grande repositorio de software en formato RPM. Aquí seguro encontrarán el paquete que están buscando.

GAME TOME: www.happypenguin.org/news



Uno de los más completos catálogos de juegos para Linux.

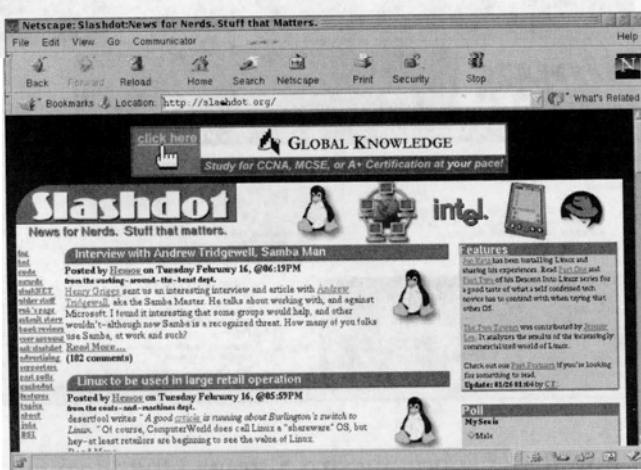
LINUX GAMES: www.linuxgames.com/



Otro sitio dedicado a Linux y los juegos.

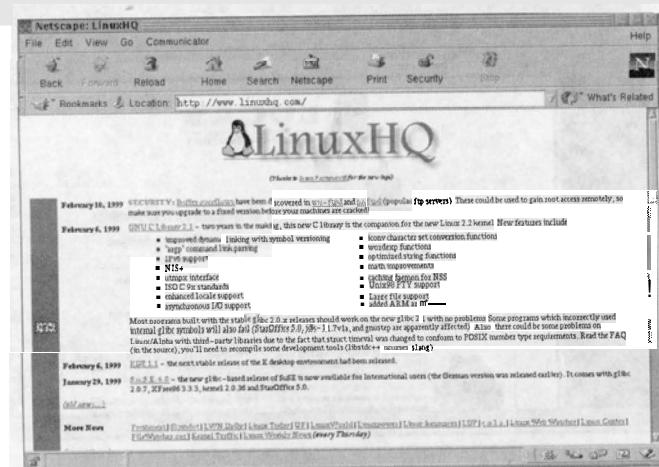
Noticias

SLASHDOT: slashdot.org/



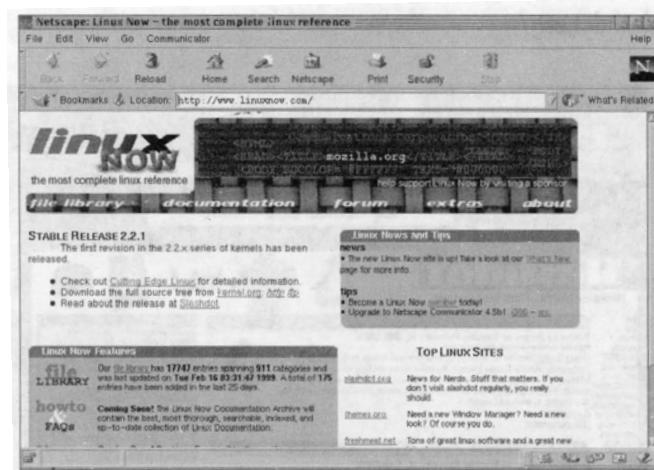
Las últimas noticias sobre Linux están aquí.

LINUXHQ: www.linuxhq.com/



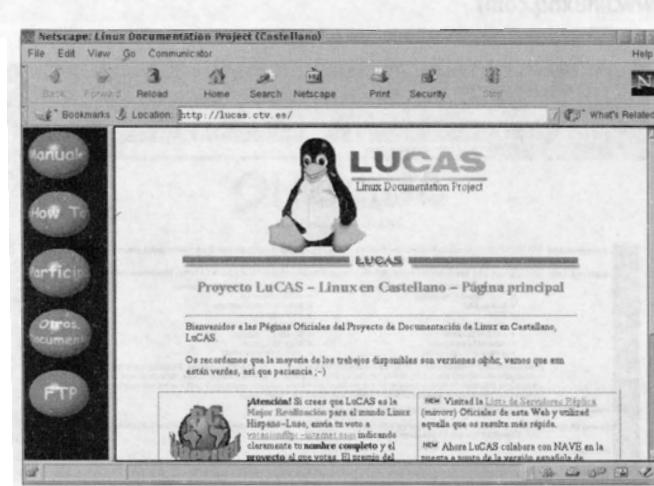
Cuartel general de Linux: noticias, novedades y la última versión del Kernel.

LINUX NOW: www.linuxnow.com/

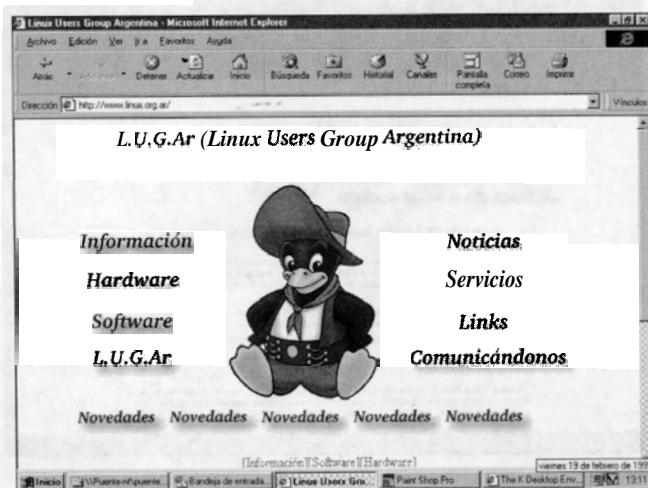


Más noticias y novedades sobre nuestro Sistema Operativo favorito.
En castellano

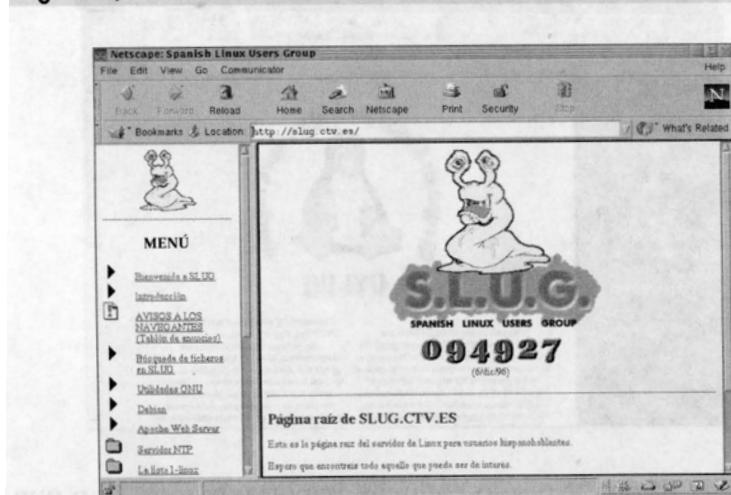
LUCAS: lucas.ctv.es



Proyecto de Documentación de Linux en Castellano. Aquí están las traducciones de los más importantes manuales de Linux. También encontraremos archivos Como, FAQ, etc.

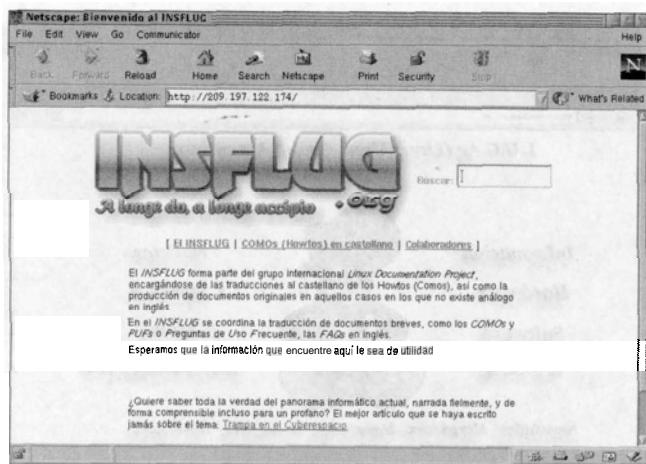
LINUX ARGENTINA: www.linux.org.ar

Ésta es la página del grupo de usuarios de Linux de Argentina. Es un buen punto de encuentro para los linuxeros del país, también encontrarán documentación.

SLUG: slug.ctv.es/

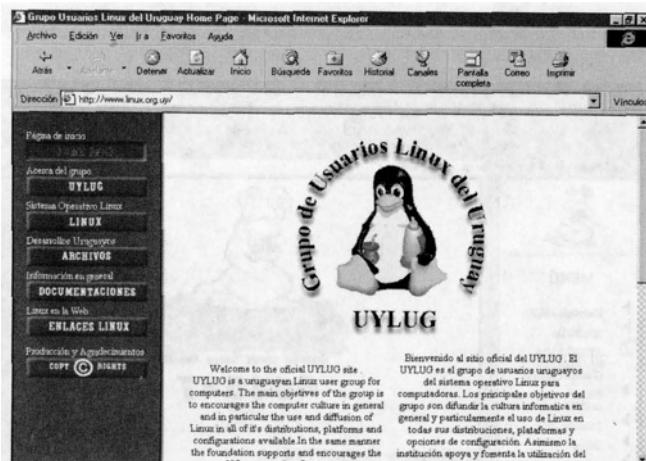
Página de los Usuarios Españoles de Linux, con abundante información y buenos enlaces.

INSLUG: 209.197.122.174/



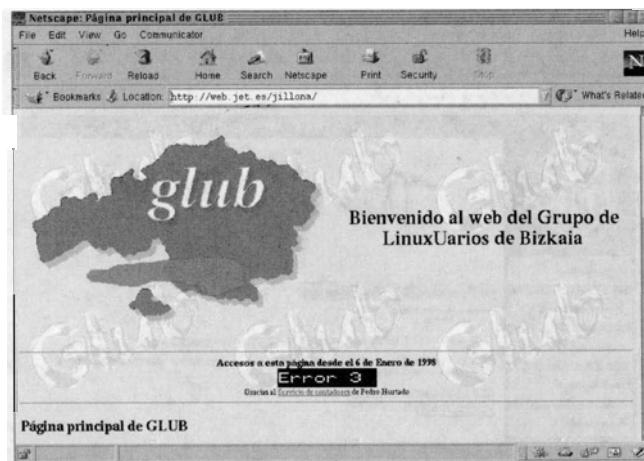
La página encargada de la traducción de los archivos COMO (howto). Visita obligada.

GRUPO DE USUARIOS DE LINUX DE URUGUAY: www.linux.org.uy/



Una muy buena página. No sólo contiene importante info, sino que cuenta con el aporte de las realizaciones de los programadores del vecino país.

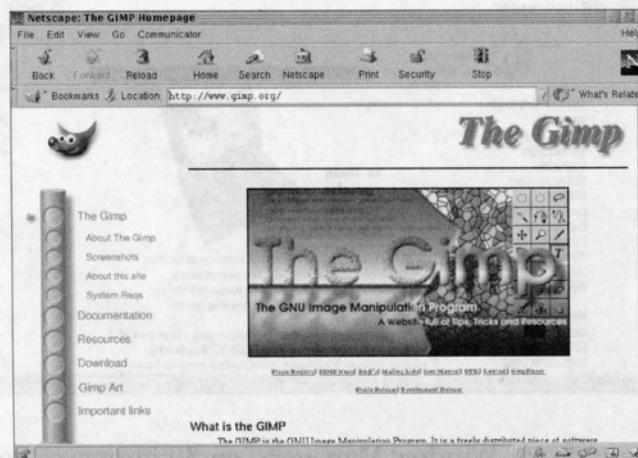
GLUB: web.jet.es/jillona/



Grupo de Linuxuarios de Bizkaia. Lo mejor de este sitio son sus links a recursos de Linux en castellano.

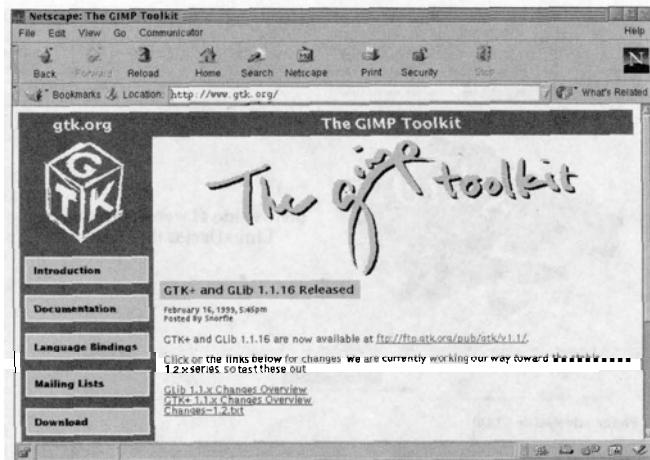
Varios

GIMP: www.gimp.org



Uno de los mejores programas de tratamiento de imágenes, imperdible.

GTK: www.gtk.org



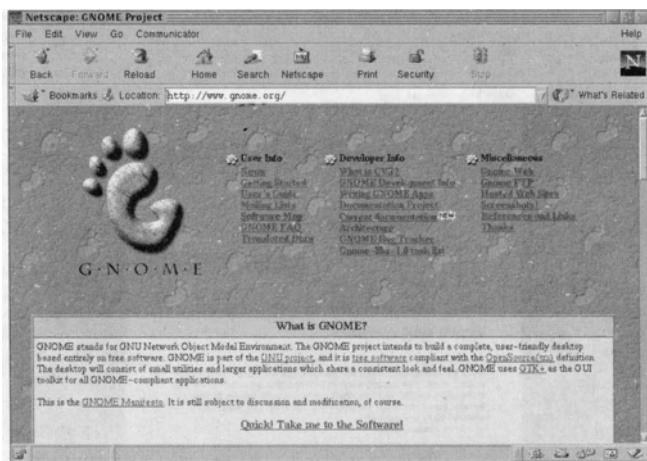
Gimp Tool Kit es la librería con la que se hizo GIMP, que actualmente es una de las más usadas para el desarrollo de aplicaciones X.

KDE: www.kde.org



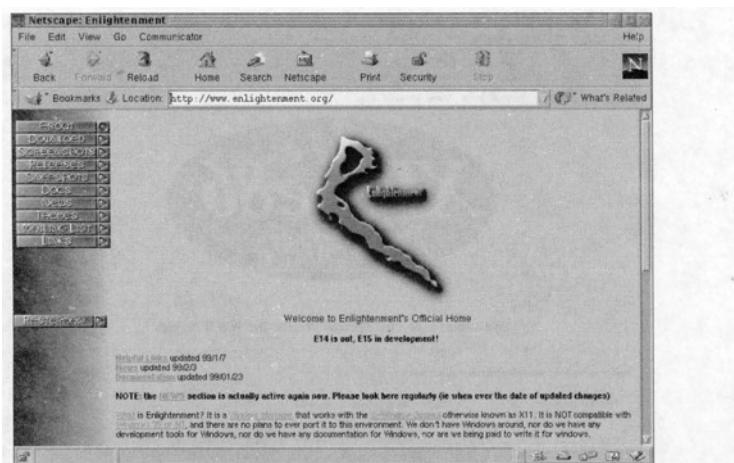
El más completo de los Window Managers, en 30 idiomas y con una buena colección de utilidades. Usa la librería QT.

GNOME: www.gnome.org



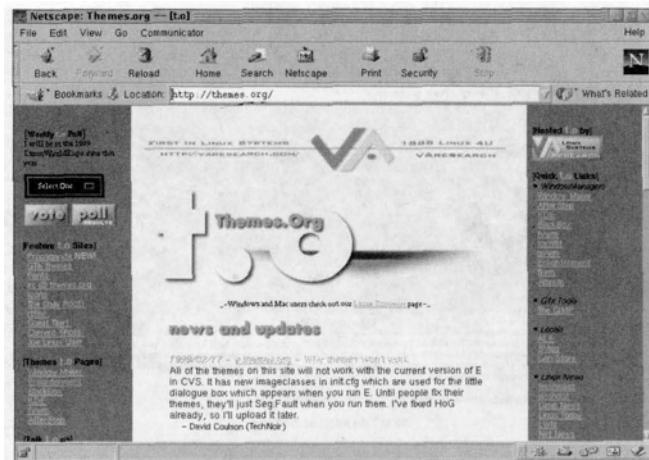
Una buena colección de utilidades basadas en la librería GTK.

ENLIGHTENMENT: www.enlightenment.org/



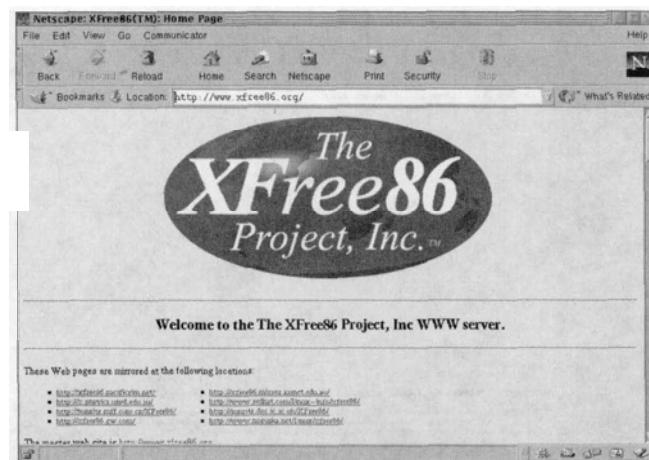
El más fascinante de los Window Managers.

THEMES: themes.org



El repositorio de todos los temas para todos los Windows Managers de Linux.

XFREE86: www.xfree86.org



La implementación de XWindow para LINUX.

ACERCA DEL CD-ROM

Apéndice

E

Acerca del CD-ROM

En el CD-ROM que acompaña este libro, ofrecemos el disco de instalación de Red Hat 5.2, la última versión de esta popular distribución. Incluye el Kernel 2.0.36, el Netscape Communicator 4.07, todos los servidores necesarios para montar un servidor de servicios de Internet, software de gestión y administración del sistema, juegos, herramientas de desarrollo, compiladores, intérpretes, libreras, LaTeX, GIMP 1.0, etc.

Tiene una pequeña variación en cuanto al proceso de instalación descripto en el Capítulo 3, que estaba referido a la versión 5.1 de la distribución. Esta variación es el agregado de la posibilidad de optar por instalaciones estándar preconfiguradas, entre las que encontraremos custom. El proceso descripto en el capítulo antes citado se adecua a esta opción; sin embargo, para quien lo instala por primera vez, esto significa una ventaja, ya que lo eximirá de la tediosa tarea de selección individual de paquetes.

Al momento de terminar con este libro, la Última versión estable del Kernel de Linux es la 2.2.1, que es la versión inmediatamente posterior a la incluida en el CD. Para actualizar nuestra instalación, sólo deberemos bajar los fuentes de esta última versión de www.kernel.org y compilarlo como se describe en el Capítulo 10 del libro.

Bibliografía

Apunte Unix Avanzado. Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile. Julio de 1998.

Linux: Instalación y primeros pasos. Matt Welsh, 1996. Traducción: Proyecto Lucas.

La Guía del enROOTador para Linux V. 2.6. Eric Dumas, diciembre de 1996. Traducción: Andrés Ramírez.

Guía de Administración de Redes con Linux. Olaf Kirch. Traducción: Proyecto Lucas.

Guía de Linux para el Usuario. Larry Greenfield. Traducción: Proyecto Lucas.

Manual de UNIX. Rev. 2.4. Jonathan Noel Tombs y Jorge Chávez Orzaez. Noviembre de 1995.

Curso práctico de Linux. Claus Denk. Universidad de Sevilla, diciembre de 1995.

Curso práctico de Linux II. Claus Denk. Universidad de Sevilla, marzo de 1997.

Using LINUX Special Edition. Jack Tackett, David Gunter y Lance Brown, 1996.

Construya un site perfecto de Internet con LINUX. LeBlanc.

índice alfabético

| | | | |
|--------------------|-----------------------|---------------------|--------------------|
| adduser | 82 | echo | 79 |
| AfterStep | 155, 153, 156 | elf | 124 |
| at | 78 | Enlightenment | 156 |
| AT&T | 16 | env | 63 |
| Ataques | 169, 170 | Ethernet | 139, 141 |
| Backdoor | 170 | exec | 102 |
| Background | 103 | expr | 79 |
| Bacteria | 171 | Fdisk | 39 |
| bg | 104, 108 | fg | 104, 106, 108 |
| bitchx | 106, 114 | File | 74 |
| BlackBox | 159, 160 | Filesystem | 130, 131 |
| Bomba lógica | 170 | Find | 78 |
| cal | 80 | FIPS | 35 |
| cat | 72 | foreground | 103 |
| cd | 72 | fork | 101, 102 |
| CD-ROM | 34, 88, 126, 130, 131 | FVWM95 | 153, 154 |
| chgrp | 78 | GID | 53, 55 |
| chmod | 72, 73, 94, 95, 96 | GNU | 18, 27, 114, 187 |
| comm | 73 | grep | 74 |
| Compilación | 114 | Grupos | 55 |
| cpt | 73, 74 | Gusano | 171 |
| Cuentas | 59 | gzip | 111 |
| date | 81, 180 | Hackers | 18 |
| deb | 111 | Hora | 45 |
| Deluser | 54 | Hostname | 143 |
| deluser | 82 | Hosts | 143 |
| Demonios | 46 | Impresora | 46, 132 |
| diff | 74 | inetd | 142, 143 |
| Disco rígido | 32 | init | 60 |
| Disk Druid | 39 | inittab | 60 |
| Dispositivos | 26 | i-nodo | 87 |
| DNS | 35, 149 | Intrusos | 168, 169, 172, 176 |
| dpkg | 102 | ISO 9660 | 27, 88, 131 |

| | | | |
|--------------------|----------------------------|-------------------|-------------------------|
| ISO8859- | 131 | Password | 47, 52, 53, 61, 82, 172 |
| jobs | 105, 108 | pci | 124 |
| KDE | 153, 168 | Permisos | 91 |
| Kernel | 19, 25, 119, 120, 121, 142 | PPP | 27, 129, 139 |
| kill | 80, 105 | pr | 76 |
| Lilo | 47, 48, 134, 135 | Proceso | 99, 106 |
| Linus Torvalds | 15 | Profile | 67 |
| ln | 74 | Prompt | 64 |
| Login | 61 | ps | 81, 99, 100 |
| lpr | 75 | pwd | 82 |
| ls | 64, 75 | Red | 34, 45 |
| mail | 82 | Red Hat | 31 |
| make | 116, 121 | rm | 76 |
| man | 80 | rmdir | 77 |
| mesg | 83 | Root | 47, 61, 112 |
| Metro-X | 44 | Routing | 144, 145, 146 |
| Midnight Commander | 66 | rpm | 31, 42, 112, 113 |
| Miguel de Icaza | 66 | Samba | 46 |
| mkdir | 75 | SCSI | 127, 128 |
| Módulos | 123 | Seguridad | 165, 166, 167 |
| Monitor | 44 | Sesión | 59 |
| More | 75, 76 | Setup | 60 |
| mount | 88 | Shadow password | 53 |
| Mouse | 43, 132, 152 | Shell | 62, 63 |
| Multics | 16 | Sistema Operativo | 19 |
| Multitarea | 27 | Sort | 77 |
| Multiusuario | 27 | Sound | 132, 133 |
| mv | 76 | Swap | 39, 40, 41, 42 |
| netcfg | 144 | tar | 111 |
| Network | 124, 127, 128, 129 | tarea | 103, 106 |
| NFS | 27, 89, 132, 139 | TCP/IP | 28, 140, 141, 142 |
| Paquetes | 42, 111 | Terminal | 59, 62, 160 |
| Particiones | 35, 39 | tgz | 111 |
| passwd | 82, 177 | Troyano | 171 |
| | | tty | 82 |
| | | UID | 53 |

| | |
|-------------------|---------------|
| Unix | 16 |
| userdel | 82 |
| Usuarios | 51, 54 |
| Variables | 63 |
| Video | 44 |
| Virus | 171 |
| | |
| wc | 77, 78 |
| who | 83 |
| WindowMaker | 153, 157, 158 |
| write | 83 |
| | |
| xfree | 86, 152, 153 |
| XWindow | 44, 151, 161 |
| zImage | 133, 134 |