Linux: Instalación y Primeros Pasos

Versión 2.2.2 - En castellano ver1.1, 8 Agosto de 1998.
Fecha de montaje: 17 de septiembre de 1998

Este es un libro de instalación y guía para nuevos usuarios del sistema Linux, dirigido tanto a los más nóveles en UNIX, como a los más expertos. Contiene información sobre como conseguir el Linux, la instalación de nuevo software, un tutorial para principiantes de UNIX y una introducción a la administración del sistema. Hemos pretendido ser tan genéricos como nos ha sido posible de tal modo que el libro pueda ser aplicable a cualquiera de las distribuciones de software para Linux.

Este libro es de distribución gratuíta. Esto quiere decir que puedes copiarlo y redistribuirlo pero bajo determinadas condiciones. Por favor, mira el copyright y las notas sobre su distribución en la página xiv.

Índice General

Prefacio

"You are in a maze of twisty little passages, all alike."

Tiene ante usted a uno de los sistemas más completos y amenazantes que jamás se ha escrito: Linux, el clónico de UNIX gratuito para el ordenador personal (PC), generado por un heterogeneo equipo formado por gurús de UNIX y hackers junto al recién llegado de turno. El sistema en sí refleja esta compleja herencia y, a pesar de que el desarrollo de Linux puede parecer un esfuerzo voluntario desorganizado, el sistema es potente, rápido y gratuito. Es un verdadero sistema operativo de 32-bits[1].

Mis propias experiencias con Linux comenzaron hace años, cuando me senté a adivinar como instalar la única "distribución" disponible en aquel tiempo—un par de disquetes que me hizo llegar H.J. Lu. Descargué una pila de ficheros y leí páginas y páginas de notas de instalación líbremente organizadas. De algún modo, conseguí instalar este sistema básico y hacer que todo funcionase. Esto fue mucho antes de que se pudiera comprar el software de Linux en CD-ROM a distribuidores mundiales; de hecho, antes de que Linux siquiera supiese como acceder a una unidad CD-ROM. Esto fue antes del XFree86, antes de Emacs, antes del soporte de software comercial, y antes de que Linux llegase a ser un verdadrero rival de MS-DOS, Microsoft Windows, y OS/2 en el mercado del ordenador personal.

Tiene en sus propias manos el mapa y libro de ruta hacia el mundo de Linux. Espero que este libro le ayude a echarse a andar con el que yo considero que es el más rápido y más potente sistema operativo que existe para el ordenador personal. Instalar su propio sistema Linux puede ser un gran momento de diversión—de modo que coja una taza de café, sientese cómodamente, y siga leyendo.

Mientras esté con ello coja una taza también para mí. Yo también he estado despierto enredando con el Linux durante días.

Audiencia

Este libro esta destinado a cualquier usuario de PC que quiera instalar y usar el Linux en su sistema. Se asume que Ud. tiene unos conocimientos básicos sobre ordenadores personales y sistemas operativos tales como MS-DOS, pero no se asume ningún conocimiento previo sobre Linux o UNIX.

[1] N. del T.: En la actualidad ya existen versiones de 64-bits para algunas otras plataformas hardware distintas del PC

A pesar de esto, sugerimos a los principiantes de UNIX que investiguen en alguno de los muchos y buenos libros que existen sobre él. Algunos de ellos aparecen listados en el Apéndice A.

Organización del Libro

Este libro contiene los siguientes capítulos:

Capítulo 1, *Introducción a Linux*, da una introducción general a lo que es Linux, que puede hacer por Ud., y que se necesita para hacerlo funcionar en su sistema. También proporciona sugerencias útiles para obtener ayuda y reducir el stress general.

Capítulo 2, *Obtener e Instalar Linux*, explica como obtener el software de Linux, así como la forma instalarlo—desde reparticionar su unidad, crear sistemas de ficheros, y cargar el software en el sistema. Contiene instrucciones que se suponen generales para cualquier distribución de Linux, y confía en que la documentación suministrada para su versión particular cubra todos los huecos.

Capítulo 3, *Tutorial de Linux*, es una introducción completa de como usar el sistema Linux para principiantes de UNIX. Si ya tiene experiencia previa con UNIX, la mayoría de este material debería serle familiar.

Capítulo 4, *Administración del Sistema*, presenta muchos de los conceptos importantes de la administración del sistema bajo Linux. Esto será también de interés para administradores de sistemas UNIX que quieran saber acerca de las formas concretas de llevar un sistema en Linux.

Capítulo 5, *Características Advanzadas*, presenta al lector un gran número de las características avanzadas que soporta el Linux, tales como el Sistema X-Window y el protocolo de red TCP/IP. Se incluye una guía completa de configuración de XFree86-3.1.

Apéndice A, *Fuentes de Información de Linux*, es un listado de otras fuentes de información acerca de Linux, incluyendo grupos de noticias, listas de distribución, documentos en-línea, y libros.

Apéndice B, *Lista de Vendedores de Linux*, proporciona una breve lista de vendedores de software que ofrecen software y servicios de Linux.

Apéndice C, *Tutorial de FTP y Lista de Sites*, es un tutorial para recibir ficheros desde Internet con FTP. Este apéndice también incluye una lista de sites de FTP que mantienen software Linux.

Apéndice D, *Lista de BBS Linux*, es una lista de BBS[2] a lo ancho del mundo que mantienen software de Linux. Dado que la mayoría de usuarios de Linux no disponen de acceso a Internet, es importante que se ponga a disposición la información sobre los sistemas BBS.

Apéndice F, *La licencia GNU General Public License*, contiene una copia de la Licencia GNU, el acuerdo de licencia bajo el cual se distribuye Linux. Es muy importante que los usuarios de Linux entiendan la GPL; en los últimos meses han surgido muchos desacuerdos sobre los términos de dicha Licencia[3].

[2] Sistema de Tablón de Boletines, Bulletin Board Sistem

[3] N. del T.: Precisamente por eso hemos decidido dejar el original de la GPL intacto y en inglés. No conocemos de la existencia de ninguna traducción oficial de la GPL al castellano.

Apéndice E, *Linux en España*, es una relación de todos aquellos aspectos de Linux que puedan interesar al usuario español. Contempla forums de mensajería españoles, BBS, canales de distribución, ftp-sites españoles, etc

Reconocimientos

Este libro ha tardado mucho en hacerse, y mucha gente ha sido responsable de su realización. En particular, me gustaría agradecer a Larry Greenfield y Karl Fogel por su trabajo en la primera versión del Capítulo 3, y a Lasr Wirzenius por su trabajo en el Capítulo 4. Gracias a Michael K. Johnson por su apoyo al LDP[4] y las convenciones LaTeX usadas en este manual, y a Ed Chi, quien me envió una copia impresa del libro para su edición.

Gracias a Melinda A. McBride de SSC, Inc., quien hizo un excelente trabajo completando el índice de los Capítulos 3, 4, y 5. También quisiera agradecer a Andy Oram, Lar Kaufman, y Bill Hahn de O'Reilly and Associates por su ayuda al Linux Documentation Project.

Gracias a Linux Systems Labs, Morse Telecommunications, Yggdrasil Computing, y otros por su soporte del Linux Documentation Project a través tanto de las ventas de este libro como de otros trabajos.

Muchas gracias a gran cantidad de activistas, incluyendo (sin ningún orden específico) a Linus Torvalds, Donald Becker, Alan Cox, Remy Card, Ted T'so, H.J. Lu, Ross Biro, Drew Eckhardt, Ed Carp, Eric Youngdale, Fred van Kempen, Steven Tweedie, y otros muchos más, por dedicar tanto tiempo y energía a este proyecto, y sin los cuales no habría nada sobre lo que escribir.

Gracias especialmente al gran número de lectores que han enviado sus útiles comentarios y correcciones. Hay tantísimos que citar aqui. ¿Quién necesita un corrector ortográfico teniendo tanta audiencia?

Matt Welsh
13 de Enero de 1994

La traducción

Lo anterior es por parte del autor. Así que, por lo que al equipo de traducción respecta tengo que agradecer especialmente a aquellos que han colaborado en traducir o revisar este gran montón de líneas, que yo solo no me hubiera atrevido ni siquiera a intentarlo: Gerardo Izquierdo, Juan Jose Amor, Eduardo Lluna, Luis Ramón Duarte, Guillermo Bautista, y Carlos Martínez Chacartegui, así como a todos aquellos que también se ofrecieron a colaborar en esta tarea.

Quiero hacer constar que todo nuestro esfuerzo ha sido llevado a cabo de una forma completamente altruista. Ninguno de nosotros somos profesionales de la traducción y eso, lamentablemente, se nota. Te ruego que seas benevolente con nosotros y que, en la medida de tus posibilidades, nos ayudes a hacer de este libro la mejor fuente de consulta en español sobre Linux.

[4]LDP es Linux Documentation Project, o Proyecto de Documentación de Linux

Un agradecimiento especial a mi gran amigo Ramón Gutierrez, quien me dio la idea de traducir este libro y me apoyó hasta terminarlo, además de encargarse de la tarea más importante, la de montar todos los trozos.

Alfonso Belloso
Agosto de 1996

El proyecto LuCAS

Este libro que tiene ante usted es el trabajo que motivó la puesta en marcha del *Proyecto LuCAS*. Dos años después de publicar la guía LIPP, LuCAS ha crecido mucho y se ha convertido en la mayor base de conocimiento que existe en castellano para el mejor sistema operativo[5].

Este éxito ha tenido lugar, al igual que el de este libro, gracias a todos los que han participado en mayor o menor medida, traduciendo o revisando, o simplemente haciéndonos llegar pequeñas pero importantes sugerencias.

Esta edición de la guía LIPP abre un proceso de puesta al día del manual de introducción a Linux (en dos años ha evolucionado mucho) al tiempo que incorpora las correcciones sugeridas por Gonzalo Daniel Molina, a quien agradecemos su colaboración desde aquí.

Juan José Amor
Julio de 1998

Créditos y aspectos Legales

El Proyecto de Documentación de Linux es un amplio grupo de escritores, correctores, y editores que están trabajando en un conjunto definitivo de manuales de Linux. El coordinador general del proyecto es Matt Welsh, ayudado por Lars Wirzenius y Michael K. Johnson.

Este manual no es más que uno entre un conjunto de varios distribuidos por el Proyecto de Documentación de Linux, que incluyen una Guía de Usuario de Linux, Guía del Administrador del Sistema, y Guía del Hacker del Kernel. Estos manuales están todos disponibles en formato fuente LaTeX y salida Postscript a través de FTP anónimo en `sunsite.unc.edu`, en el directorio `/pub/Linux/docs/LDP`.[6]

Animamos a cualquiera al que le guste escribir o editar a que se una a nosotros a fin de mejorar la documentación de Linux. Si dispone de acceso a correo electrónico de Internet (e-mail), puede apuntarse al canal `DOC` de la lista de distribución de correo `Linux-Activists` enviando un mensaje a

`linux-activists-request@niksula.hut.fi`

[5] Todos los trabajos de LuCAS pueden accederse, gratuitamente, a través de la dirección Internet `http://lucas.ctv.es/`

[6] N. del T.: Esperamos que, en breve, pueda recoger esta versión traducida y actualizada también allí

con la línea

```
X-Mn-Admin: join DOC
```

como primera línea del cuerpo del mensaje.

Sírvase ponerse en contacto con el autor y coordinador de este manual si tiene preguntas, postales, dinero, o ideas. Matt Welsh puede ser localizado via Internet e-mail en `mdw@sunsite.unc.edu`, y en la vida real en

205 Gray Street
Wilson, N.C. 27896
U.S.A.

205 Gray Street NE, Wilson NC, 27893 USA
`mdw@sunsite.unc.edu`

[7] N. del T.: Las mismas condiciones son aplicables a esta traducción

Se pueden conceder excepciones a estas reglas con fines académicos: Escriba a Matt Welsh, a la dirección de arriba, o por correo electrónico a `mdw@sunsite.unc.edu`, y pregunte. Estas restricciones están aquí para protegernos como autores, no para restringirles a Uds. como educadores o aprendices.

El autor anima a los distribuidores de software de Linux en cualquier medio a usar el libro como una guía de instalación y de nuevo usuario. Dado el copyright de arriba, Ud. es libre de imprimir y distribuir copias de este libro con su software. Puede optar por distribuir este libro sin cargo alguno, o para lograr un beneficio. Si lo hace asi, puede desear incluir un pequeño “suplemento de instalación” para su versión.

El autor desearía saber de cualquier plan de publicar y distribuir este libro comercialmente. De este modo, podemos asegurar de que Ud. se mantiene al día con nuevas revisiones. Y, si hubiese una nueva versión a punto de salir, Ud. podría desear retrasar la publicación del libro hasta que esté disponible.

Si Ud. esta distribuyendo este libro comercialmente, cualquier donación, royalties, y/o copias impresas serían enormemente apreciadas por el autor. La contribución en este sentido muestra su apoyo al software libre y al Proyecto de Documentación de Linux.

Todo el código fuente en *Linux Installation and Getting Started* se ha puesto bajo la licencia GNU General Public License. Vea el Apéndice F con una copia de la licencia GNU “GPL.”

Notación usada en el documento

Este acuerdo de notación debería de ser obvio, no obstante se incluye aquí por si acaso.

Negita Usado para resaltar **conceptos nuevos**, **AVISOS**, y **palabras clave** de un lenguaje.

itálicas Usado para *enfatizar* en el texto, y ocasionalmente para citas o presentaciones al comienzo de una sección. También usado para indicar comandos que teclee el usuario cuando se muestra interacción con la pantalla (ver debajo).

⟨*sesgado*⟩ Usado para marcar **meta-variables** en el texto, especialmente en representaciones de la línea de comandos. Por ejemplo,

`ls -l` ⟨*foo*⟩

donde ⟨*foo*⟩ “equivale a” un nombre de fichero, como `/bin/cp`.

`Escritura de máquina`
Usada para representar interacción de pantalla, como en

```
$ ls -l /bin/cp
-rwxr-xr-x  1 root    wheel    12104 Sep 25 15:53 /bin/cp
```

También se usa en ejemplos de código, ya sea código C, un script de la shell, o cualquier otro, y para mostrar ficheros en general, tales como ficheros de configuración. Cuando sea necesario, y por una mejor claridad, estos ejemplos o figuras se incluirán en cajas.

`Tecla` Representa una tecla a pulsar. A menudo lo verá de esta forma:

> Press return to continue.

◇ Un diamante en el margen, como un diamante negro en una pista de esquí, señala "peligro" o "precaución". Lea detenidamente los párrafos marcados de esta forma.

Capítulo 1

Introducción a Linux

Linux es probablemente el acontecimiento más importante del software gratuito desde el original *Space War*, o, más recientemente, Emacs. Se ha convertido en el sistema operativo para los negocios, educación, y provecho personal. Linux ya no es solo para gurús de UNIX que se sientan durante horas frente a la resplandeciente consola (aunque le aseguramos que un gran número de usuarios pertenece a esta categoría). Este libro le ayudará a sacarle el máximo partido.

Linux (pronounciado con una *i* corta, como en *LIH-nucs*) es un clónico del sistema operativo UNIX que corre en ordenadores Intel 80386 y 80486. Soporta un amplio rango de software, desde TEX a X Windows al compilador GNU C/C++ a TCP/IP. Es una implementación de UNIX versátil, distribuida gratuitamente en los términos de la Licencia GNU (vea el Apéndice F).

Linux puede convertir cualquier PC 386 o 486 en una estación de trabajo. Le pondrá todo el poder de UNIX en la punta de sus dedos. En los negocios ya se instala Linux en redes enteras, usando el sistema operativo para manejar registros financieros y de hospitales, un entorno de usuario distribuido, telecomunicaciones, etc. Universidades de todo el mundo usan Linux para dar cursos de programación y diseño de sistemas operativos. Y, por supuesto, entusiastas de los ordenadores de todo el mundo están usando Linux en casa, para programar, entretenerse, y conocerlo a fondo.

Lo que hace a Linux tan diferente es que es una implementación *gratuita* de UNIX. Fue y aun es desarrollado por un grupo de voluntarios, principalmente en Internet, intercambiando código, comentando fallos, y arreglando los problemas en un entorno abierto. Cualquiera es bienvenido a sumarse al esfuerzo de desarrollo de Linux: todo lo que se pide es interés en producir un clónico gratuito de UNIX y algunos conocimientos de programación. El libro que tiene en sus manos es su guía de viaje.

1.1 Sobre este libro

Este libro es una guía de instalación e iniciación al sistema Linux. El objetivo es conseguir que los nuevos usuarios se adapten y hagan funcionar el sistema, condensando tanto material importante como sea posible dentro de un libro. En lugar de abarcar muchos de los volátiles detalles técnicos, esas cosas que tienden a cambiar con el veloz desarrollo, le ofrecemos una base suficiente con la que podrá buscar más por Ud. mismo.

Linux no es difícil de instalar y usar. Sin embargo, como cualquier implementación de UNIX, a menudo hay algo de magia negra implicada para conseguir que todo funcione correctamente. Esperamos que este libro le introduzca en el mundo del Linux y le muestre cuan atractivo puede llegar a ser este sistema operativo.

En este libro se tratan los siguientes temas.

- ¿Qué es Linux? El diseño y la filosofía de este sistema operativo único, y lo que puede hacer por Ud.
- Todos los detalles de lo que es necesario para correr Linux, incluyendo sugerencias sobre que tipo de configuración hardware se recomienda para un sistema completo.
- Como obtener e instalar Linux. Hay muchas distribuciones del software de Linux. Presentamos una discusión general de las distribuciones de software de Linux, como obtenerlas, e instrucciones genéricas para instalar el software (que deberían ser aplicables a cualquier distribución).

  Esta edición también contiene instrucciones específicas para la distribución Slackware de Linux.
- Un breve tutorial de introducción a UNIX, para aquellos usuarios que nunca antes hayan tenido contacto con UNIX. Este tutorial debería proporcionar, esperamos, material suficiente para que completos novatos consigan los conocimientos básicos de como moverse por el sistema.
- Una introducción a la administración de sistemas con Linux. Esto abarca las tareas más importantes con las que los nuevos administradores de Linux necesitarán familiarizarse, tales como crear usuarios, manejar los sistemas de ficheros, y muchas más.
- Información sobre configuración de aspectos más avanzados de Linux, como el Sistema X Window, redes con TCP/IP y SLIP, y la puesta en marcha del correo electrónico y sistemas de news.

Este libro es para el usuario de ordenador personal que desea iniciarse en Linux. No se asume experiencia previa de UNIX, pero se espera que los novatos busquen más materiales sobre la marcha. Para los no familiarizados con UNIX, se da una lista de fuentes de información útiles en el Apéndice A. En general, se pretende que este libro se lea junto con otro libro sobre conceptos básicos de UNIX.

1.2 Breve historia de Linux

UNIX es uno de los sistemas operativos más populares del mundo debido a su extenso soporte y distribución. Originalmente fue desarrollado como sistema multitarea con tiempo compartido para miniordenadores y *mainframes* a mediados de los 70, y desde entonces se ha convertido en uno de los sistemas más utilizados a pesar de su, ocasionalmente, confusa interfaz con el usuario y el problema de su estandarización.

¿Cuál es la verdadera razón de la popularidad de UNIX? Muchos hackers consideran que UNIX es el auténtico y único sistema operativo. El desarrollo de Linux parte de un grupo en expansión de hackers de UNIX que quisieron hacer su sistema con sus propias manos.

Existen numerosas versiones de UNIX para muchos sistemas, desde ordenadores personales hasta supercomputadores como el Cray Y-MP. La mayoría de las versiones de UNIX para ordenadores personales son muy caras. Cuando se escribía este libro, una copia para una máquina 386 del UNIX System V de AT&T costaba unos 1500 dólares estadounidenses.

Linux es una versión de UNIX de libre distribución, inicialmente desarrollada por Linus Torvalds[1] en la Universidad de Helsinki, en Finlandia. Fue desarrollado con la ayuda de muchos programadores y expertos de Unix a lo largo y ancho del mundo, gracias a la presencia de Internet. Cualquier habitante del planeta puede acceder a Linux y desarrollar nuevos módulos o cambiarlo a su antojo. El núcleo de Linux no utiliza ni una sola línea del código de AT&T o de cualquier otra fuente de propiedad comercial, y buena parte del software para Linux se desarrolla bajo las reglas del proyecto de GNU de la Free Software Foundation, Cambridge, Massachusetts.

Inicialmente, sólo fue un proyecto de aficiónado de Linus Torvalds. Se inspiraba en Minix, un pequeño UNIX desarrollado por Andy Tanenbaum, y las primeras discusiones sobre Linux surgieron en el grupo de News `comp.os.minix`. Estas discusiones giraban en torno al desarrollo de un pequeño sistema UNIX de carácter académico dirigido a aquellos usuarios de Minix que querían algo más.

El desarrollo inicial de Linux ya aprovechaba las características de conmutación de tareas en modo protegido del 386, y se escribió todo en ensamblador. Linus dice,

> "Comencé a utilizar el C tras escribir algunos drivers, y ciertamente se aceleró el desarrollo. En este punto sentí que mi idea de hacer un 'un Minix mejor que Minix' se hacía más seria. Esperaba que algún día pudiese recompilar el `gcc` bajo Linux...
>
> "Dos meses de trabajo, hasta que tuve un driver de discos (con numerosos *bugs*, pero que parecía funcionar en mi PC) y un pequeño sistema de ficheros. Aquí tenía ya la versión 0.01 [al final de Agosto de 1991]: no era muy agradable de usar sin el driver de disquetes, y no hacía gran cosa. No pensé que alguien compilaría esa versión."

No se anunció nada sobre esa versión, puesto que las fuentes del 0.01 jamás fueron ejecutables: contenían solo rudimentos de lo que sería el núcleo , y se asumía que se tenía acceso a un Minix para poderlo compilar y jugar con él.

El 5 de Octubre de 1991, Linus anunció la primera versión "oficial" de Linux, la 0.02. Ya podía ejecutar `bash` (el shell de GNU) y `gcc` (el compilador de C de GNU), pero no hacía mucho más. La intención era ser un juguete para hackers. No había nada sobre soporte a usuarios, distribuciones, documentación ni nada parecido. Hoy, la comunidad de Linux aun trata estos asuntos de forma secundaria. Lo primero sigue siendo el desarrollo del *kernel*.

Linus escribía en `comp.os.minix`,

> "¿Suspiráis al recordar aquellos días de Minix-1.1, cuando los hombres eran hombres y escribían sus propios drivers? ¿Os sentís sin ningún proyecto interesante y os gustaría

[1] `torvalds@kruuna.helsinki.fi`.

> tener un verdadero S.O. que pudiérais modificar a placer? ¿Os resulta frustrante el tener solo a Minix? Entonces, este artículo es para vosotros.
>
> "Como dije hace un mes, estoy trabajando en una versión gratuita de algo parecido a Minix para ordenadores At-386. He alcanzado la etapa en la que puede ser utilizable y voy a poner las fuentes para su distribución. Es solo la versión 0.02... pero he conseguido ejecutar en él `bash`, `gcc`, `gnu-make`, `gnu-sed`, `compress`, etc.

Tras la versión 0.03, Linus saltó a la versión 0.10, al tiempo que más gente empezaba a participar en su desarrollo. Tras numerosas revisiones, se alcanzó la versión 0.95, reflejando la esperanza de tener lista muy pronto una versión "oficial". (Generalmente, la versión 1.0 de los programas se corresponden con la primera teóricamente completa y sin errores). Esto sucedía en Marzo de 1992. Año y medio después, en Diciembre del 93, el núcleo estaba en la revisión 0.99.pl14, en una aproximación asintótica al 1.0. Actualmente, el núcleo se encuentra en la versión 1.1 parche 52, y se acerca la 1.2.[2]

Hoy Linux es ya un clónico de UNIX completo, capaz de ejecutar X Window, TCP/IP, Emacs, UUCP y software de correo y News. Mucho software de libre distribución ha sido ya portado a Linux, y están empezando a aparecer aplicaciones comerciales. El hardware soportado es mucho mayor que en las primeras versiones del núcleo. Mucha gente ha ejecutado tests de rendimiento en sus sistemas Linux 486 y se han encontrado que son comparables a las estaciones de trabajo de gama media de Sun Microsystems y Digital. ¿Quién iba a imaginar que este "pequeño" clónico de UNIX iba a convertirse en un estándar mundial para los ordenadores personales?

1.3 Características del sistema

Linux implementa la mayor parte de las características que se encuentran en otras implementaciones de UNIX, más algunas otras que no son habituales. En esta sección nos daremos una vuelta por todo ello.

Linux es un sistema operativo completo con multitarea y multiusuario (como cualquier otra versión de UNIX). Esto significa que pueden trabajar varios usuarios simultáneamente en él, y que cada uno de ellos puede tener varios programas en ejecución.

El sistema Linux es compatible con ciertos estándares de UNIX a nivel de código fuente, incluyendo el IEEE POSIX.1, System V y BSD. Fue desarrollado buscando la portabilidad de los fuentes: encontrará que casi todo el software gratuito desarrollado para UNIX se compila en Linux sin problemas. Y todo lo que se hace para Linux (código del núcleo, drivers, librerías y programas de usuario) es de libre distribución.

En Linux también se implementa el control de trabajos POSIX (que se usa en los *shells* `csh` y `bash`), las pseudoterminales (dispositivos `pty`), y teclados nacionales mediante manejadores de teclado cargables dinámicamente. Además, soporta **consolas virtuales**, lo que permite tener más de una sesión abierta en la consola de texto y conmutar entre ellas fácilmente. A los usuarios del programa "`screen`" les resultará familiar esto.

[2] N. del T.: En el momento de traducir estas líneas la versión estable del núcleo es la 1.2.13, pero el desarrollo continúa por la 1.3.47 en versión beta ...

El núcleo es capaz de emular por su cuenta las instrucciones del coprocesador 387, con lo que en cualquier 386 con coprocesador o sin él se podrán ejecutar aplicaciones que lo requieran.

Linux soporta diversos sistemas de ficheros para guardar los datos. Algunos de ellos, como el *ext2fs*, han sido desarrollados específicamente para Linux. Otros sistemas de ficheros, como el Minix-1 o el de Xenix también están soportados. Y con el de MS-DOS se podrán acceder desde Linux a los disquetes y particiones en discos duros formateados con MS-DOS. Además, también soporta el ISO-9660, que es el estándar seguido en el formato de los CD-ROMs. Hablaremos más sobre los sistemas de ficheros en los capítulos 2 y 4.

Linux implementa todo lo necesario para trabajar en red con TCP/IP. Desde manejadores para las tarjetas de red más populares hasta SLIP/PPP, que permiten acceder a una red TCP/IP por el puerto serie. También se implementan PLIP (para comunicarse por el puerto de la impresora) y NFS (para acceso remoto a ficheros). Y también se han portado los clientes de TCP/IP, como FTP, `telnet`, NNTP y SMTP. Hablaremos más acerca de esto en el capítulo 5.

El núcleo de Linux ha sido desarrollado para utilizar las características del modo protegido de los microprocesadores 80386 y 80486. En concreto, hace uso de la gestión de memoria avanzada del modo protegido y otras características avanzadas. Cualquiera que conozca la programación del 386 en el modo protegido sabrá que este modo fue diseñado para su uso en UNIX (o tal vez Multics). Linux hace uso de esta funcionalidad precisamente.

El núcleo soporta ejecutables con paginación por demanda. Esto significa que sólo los segmentos del programa que se necesitan se cargan en memoria desde el disco. Las páginas de los ejecutables son compartidas mediante la técnica *copy-on-write*, contribuyendo todo ello a reducir la cantidad de memoria requerida para las aplicaciones.

Con el fin de incrementar la memoria disponible, Linux implementa la paginación con el disco: puede tener hasta 256 megabytes de espacio de intercambio o "swap"[3] en el disco duro. Cuando el sistema necesita más memoria, expulsará páginas inactivas al disco, permitiendo la ejecución de programas más grandes o aumentando el número de usuarios que puede atender a la vez. Sin embargo, el espacio de intercambio no puede suplir totalmente a la memoria RAM, ya que el primero es mucho más lento que ésta.

La memoria dedicada a los programas y a la *cache* de disco está unificada. Por ello, si en cierto momento hay mucha memoria libre, el tamaño de la cache de disco aumentará acelerando así los accesos.

Los ejecutables hacen uso de las librerías de enlace dinámico.

Esto significa que los ejecutables comparten el código común de las librerías en un único fichero, como sucede en SunOS. Así, los ejecutables serán más cortos a la hora de guardarlos en el disco, incluyendo aquellos que hagan uso de muchas funciones de librería. También pueden enlazarse estáticamente cuando se deseen ejecutables que no requieran la presencia de las librerías dinámicas en el sistema. El enlace dinámico se hace en tiempo de ejecución, con lo que el programador puede cambiar las librerías sin necesidad de recompilación de los ejecutables.

Para facilitar la depuración de los programas, el núcleo de Linux puede generar volcados de

[3] El nombre swap es inadecuado: no se intercambian procesos completos, sino páginas individuales. Por supuesto, en muchos casos se expulsan al disco procesos completos, pero no siempre ocurre

la imagen de memoria de los programas (ficheros `core`). Entre esto y la posibilidad de compilar ejecutables con soporte de depuración, el programador podrá averiguar la causa de los fallos de su programa.

1.4 Software

En esta sección le haremos una breve introducción sobre las muchas aplicaciones software disponibles para Linux y sobre tareas típicas con ordenadores. Después de todo la parte más importante del sistema radica en la cantidad de software disponible que existe. El hecho de que la mayor parte de ese software sea gratis lo hace aún más impresionante.

1.4.1 Comandos y utilidades básicas

Prácticamente cada utilidad que espere encontrar en un sistema UNIX estandard ha sido transladada a Linux. Esto incluye comandos básicos como `ls`, `awk`, `tr`, `sed`, `bc`, `more`, y muchos más. Diga el nombre de una. Es seguro que Linux la tiene. En Linux puede esperar encontrar un entorno de trabajo que le sera familiar si procede de entornos UNIX. Todos los comandos estandard y utilidades están ahí. (Los usuarios novatos de Linux deberían ver el Capítulo 3 para una introducción a los comandos básicos del UNIX.)

Hay disponibles numerosos editores de texto, incluyendo `vi`, `ex`, `pico`, `jove`, GNU Emacs y sus variantes como Lucid Emacs (el cual incorpora extensiones para usarlo bajo X Window), y `joe`. Sea cual sea el editor que esté acostumbrado a usar, es prácticamente seguro que habrá sido portado a Linux.

La elección de un editor de texto es un asunto interesante. Muchos usuarios de UNIX siguen usando editores "simples", como `vi` (de hecho, el autor escribió este libro usando `vi` sobre Linux. Pero `vi` tiene muchas limitaciones debido a su antigedad, por lo que están ganando popularidad editores más modernos (y complejos) como Emacs. Emacs proporciona un completo lenguaje de macros basadas en LISP con su intérprete, una poderosa sintaxis de órdenes y multitud de otras opciones interesantes. Existe un conjunto de macros de Emacs para leer correo electrónico y 'news', moverse por el árbol de directorios, e incluso tener una sesión de psicoterapia con un psicoterapeuta de inteligencia artificial (indispensable para Linuxeros estresados :-))

Un punto interesante es que la mayoría de las utilidades básicas para Linux son programas GNU. Estas utilidades GNU proporcionan características avanzadas que no se encuentran en las versiones estandard para BSD ó AT&T. Por ejemplo, la versión de GNU del editor `vi`, `elvis`, incluye un lenguaje de macros estructurado que difiere de la versión inicial de AT&T. De cualquier modo, las utilidades GNU se esfuerzan por mantenerse compatibles con sus equivalentes BSD y System V. Mucha gente considera las versiones GNU de estos programas superiores a las originales.

La utilidad más importante para la mayoría de los usuarios es el **intérprete de comandos**. El intérprete de comandos es un programa que lee y ejecuta órdenes del usuario. Además, muchas proporcionan características como **control de procesos** (permitiendo al usuario manejar varios procesos corriendo a la vez), redirección de entrada/salida, y un lenguaje de órdenes para escribir

scripts. Una **script** es un fichero que contiene un programa en el lenguaje de ordenes del intérprete de comandos, similar a los ficheros "batch" de MS-DOS.

Hay varios tipos de intérpretes de comandos disponibles para Linux. La principal diferencia entre ellos es el lenguaje de comandos. Por ejemplo, el **C Shell** (`csh`) usa un lenguaje de comandos muy parecido al lenguaje de programación C. El clásico **Bourne Shell** usa un lenguaje de comandos diferente. Un argumento para la elección de un determinado intérprete de comandos es el lenguaje de comandos que proporciona. El intérprete de comandos que use, definirá su entorno de trabajo bajo Linux.

No importa el intérprete de comandos que esté acostumbrado a usar, alguna versión de este habrá sido probablemente llevada a Linux. La más popular es el GNU Bourne Again Shell (`bash`),, que es una variante del Bourne clásico que incluye muchas características avanzadas como control de procesos, historial de órdenes, terminación de comandos y ficheros, edición de la línea de comandos al estilo Emacs y poderosas extensiones al lenguaje de comandos del Bourne clásico. Otro intérprete de comandos muy popular es el `tcsh`, una versión del C Shell con funcionalidad avanzada similar a la encontrada en `bash`. Otros intérpretes de comandos son `zsh`, un pequeño intérprete similar al Bourne; el Korn (`ksh`); BSD's `ash` y `rc`, el intérprete de comandos de Plan 9.

Linux le da la oportunidad única de configurar el sistema a su gusto según sus necesidades. Por ejemplo, si usted es la única persona que usa el sistema, y prefiere el editor `vi`, y el intérprete de comandos `bash`, no es necesario que instale otros editores e intérpretes de comandos. La actitud "hágaselo usted mismo" es la que se impone entre los usuarios de Linux.

1.4.2 Formateado y proceso de textos

Casi la totalidad de los usuarios de ordenadores tiene la necesidad de usar un sistema de preparación de documentos de algún tipo. (¿Cuantos entusiastas de los ordenadores conoces que todavía usen papel y lapiz?. No demasiados, apostamos por ello). En el mundo de los PC, *procesado de textos* es la norma: esto implica edición y manipulación de texto (a menudo en un entorno WYSIWYG [4] y la producción de copias impresas del texto con dibujos, tablas y otros adornos.

En el mundo UNIX, el *formateado de textos* es mucho más común. Es bastante diferente del clásico concepto de proceso de textos. Con un sistema de formateado de textos, el texto es introducido por el autor usando un "lenguaje de composición", que describe como debe ser formateado el texto. En lugar de introducir el texto dentro de un entorno de proceso de textos especial, los fuentes del texto pueden ser modificados con cualquier editor de textos como `vi` o Emacs. Una vez el texto fuente está completo (en el lenguaje de composición), el usuario formatea el texto con un programa separado, que convierte el texto fuente a un formato adecuado para la impresión. Esto es en cierta forma análogo a programar en un lenguaje como C, y "compilar" el documento a una forma imprimible.

Hay muchos formateadores de textos disponibles para Linux. Uno es `groff`, la versión GNU del clásico `nroff` originalmente desarrollado por Bell Labs y todavía usado en muchos sistemas UNIX por todo el mundo. Otro sistema de formateado de textos moderno es $\TeX$, desarrollado por Donald

[4] N. del T.: WYSIWYG: "What-You-See-Is-What-You-Get". "Lo-que-ves-es-lo-que-obtienes"

Knuth de gran popularidad. Dialectos de TEX, como LATEX, también están disponibles.

Formateadores de texto como TEX y `groff` difieren entre si principalmente en la sintaxis del lenguaje de composición. La elección de un sistema de formateo frente a otro estará basada en las utilidades disponibles para satisfacer sus necesidades, así como en su gusto personal.

Por ejemplo, algunas personas consideran que el lenguaje de composición de `groff` es un poco oscuro por lo que usan TEX, que es más legible para humanos. Pero, `groff` es capaz de producir salida en ASCII llano, visualizable en un terminal, mientras que TEX está destinado principalmente para salida a impresora. Por lo tanto, existen varios programas para producir salida ASCII de documentos formateados con TEX, o para convertir TEX a `groff`.

Otro sistema de formateo de texto es `texinfo`, una extensión de TEX usada para la documentación de programas por la 'Free Software Foundation'. `texinfo` es capaz de producir documentos impresos o un documento "Info" con hiperenlaces por los que nos podemos mover desde un único fichero fuente. Los ficheros Info son el formato principal de documentación usado por paquetes de GNU como Emacs.

Los formateadores de texto son usados ampliamente en la comunidad informática para producir informes, tesis, artículos de revistas y libros (de hecho, este libro ha sido producido con LATEX). La capacidad de procesar el lenguaje fuente como un fichero de texto llano abre la puerta a muchas extensiones al formateado de texto en sí, puesto que los documentos no son guardados en un oscuro formato, legible solo por un procesador de textos particular, los programadores tienen la posibilidad de escribir analizadores y traductores para el lenguaje de composición y extender el sistema.

¿Qué aspecto tiene el lenguaje de composición? En general, el texto fuente consiste principalmente en el texto en si mismo, junto con "códigos de control" para producir efectos particulares, como cambio de tipo de letra, seleccionar margenes, crear listas, ..etc.

Como ejemplo, tomemos el siguiente texto:

> Sr. Torvalds:
>
> Estamos muy preocupados con sus planes actuales de añadir *sugestión post-hipnótica* en el código del terminal de **Linux**. Nos sentimos así por tres razones:
>
> 1. Mostrar mensajes subliminares en el terminal no es solo inmoral, sino una pérdida de tiempo;
> 2. Ha sido probado que las sugestiones post-hipnóticas no son efectivas cuando se usan sobre 'UNIX hackers' desprevenidos;
> 3. Ya hemos añadido descargas eléctricas de alto voltaje como medida de seguridad en el código de `login`.
>
> Esperamos que lo reconsidere.

Este texto aparecería en el lenguaje de formateo de LATEX como sigue:

```
\begin{quote}
Sr. Torvalds:
```

```
Estamos muy preocupados con sus planes actuales de a\~nadir
{\em sugesti\'on post-hipn\'otica\/} en el c\'odigo del
terminal de {\bf Linux}. Nos sentimos as\'{\i} por tres razones:
\begin{enumerate}
\item Mostrar mensajes subliminares en el terminal no es solo
      inmoral, sino una p\'erdida de tiempo;
\item Ha sido probado que las sugestiones post-hipn\'oticas
      no son efectivas cuando se usan sobre `UNIX hackers'
      desprevenidos;
\item Ya hemos a\~nadido descargas el\'ectricas de alto voltaje
      como medida de seguridad en el c\'odigo de {\tt login}.
\end{enumerate}
Esperamos que lo reconsidere.
\end{quote}
```

El autor escribe el texto 'fuente' anterior usando un editor de texto y genera la salida formateada procesando el texto con LaTeX. A primera vista, el lenguaje de composición puede parecer oscuro, pero es realmente fácil de aprender. Usar un sistema de formateo de textos hace cumplir las normas tipográficas en la escritura. Por ejemplo, todas las listas enumeradas dentro del documento tendrán el mismo aspecto, aunque el autor modifique la definición del 'entorno' de la lista enumerada. La meta principal es permitir al autor concentrarse en la escritura del texto en lugar de preocuparse por las convenciones tipográficas.

Los procesadores de texto WYSIWYG son atractivos por muchas razones; proporcionan un poderoso (y a veces complejo) interface visual para la edición del documento. Pero este interface está inherentemente limitado por aquellos aspectos de la estructura del texto que son accesibles al usuario. Por ejemplo, muchos procesadores de texto proporcionan un 'lenguaje de formateado' especial para producir expresiones complicadas como pueden ser las fórmulas matemáticas. Esto es idéntico al formateado de textos, aunque en una escala mucho menor.

El sutil beneficio del formateado de textos es que el sistema te permite especificar exactamente lo que quieres. También, los formateadores de texto permiten editar el texto fuente con cualquier editor de texto y el fuente es fácilmente convertible a otros formatos. Esta flexibilidad y potencia la obtenemos a cambio de renunciar al interface WYSIWYG.

Muchos usuarios de procesadores de texto estan acostumbrados a ver el texto formateado tal y como lo editaron. Por otra parte, cuando se escribe con un formateador de texto, generalmente no nos preocupamos sobre el aspecto que tendrá una vez formateado. El escritor aprende a imaginarse el aspecto que tendrá el texto una vez formateado a partir de las órdenes de formateado usadas en el fuente.

Hay programas que permiten ver el documento formateado en una pantalla gráfica antes de imprimirlo. Por ejemplo, `xdvi` visualiza un fichero 'independiente de dispositivo' generado por TeX en X Windows. Otras aplicaciones, como `xfig` proporcionan interfaces gráficos WYSIWYG para dibujar, los cuales son posteriormente convertidos al lenguaje de formateo de textos para su inclusión en el documento.

Los formateadores de texto como `nroff` están disponibles desde mucho antes de que apareciesen los procesadores de texto. Sin embargo, mucha gente todavía prefiere el uso de los formateadores de texto porque son más versátiles e independientes del entorno gráfico. En cualquier caso, el procesador de textos `idoc` está también disponible en Linux y no pasará mucho tiempo hasta que también veamos procesadores de texto comerciales disponibles. Si de ninguna forma quieres renunciar al proceso de texto en favor del formateo de textos, siempre puedes ejecutar MS-DOS o algún otro sistema operativo además de Linux.

Hay disponibles muchas otras utilidades relacionadas con el formateo de textos. El potente sistema METAFONT, usado para diseñar fuentes para TEX, está incluido el la distribución de TEX para Linux. Otros programas incluyen `ispell`, un corrector ortográfico interactivo; `makeindex`, usado para generar índices en documentos LATEX; así como muchos paquetes de macros para `groff` y TEX para el formateo de diferentes tipos de documentos y textos matemáticos. Hay disponibles programas de conversión entre fuentes de TEX y `groff` a miles de otros formatos.

1.4.3 Lenguajes de programación y utilidades

Linux proporciona un completo entorno de programación UNIX, incluyendo todas las librerías estandard, herramientas de programación, compiladores, depuradores y todo aquello que esperarías encontrar en otro sistema UNIX. En el mundo del desarrollo de programas UNIX, las aplicaciones y sistemas suelen ser programados en C ó C++. El compilador estandard de C y C++ para Linux es el GNU `gcc`, el cual es un avanzado y moderno compilador que permite multitud de opciones. Es también capaz de compilar C++ (incluyendo las características de AT&T 3.0) así como en Objetive-C y otros dialectos de C orientados a objetos.

Además de C y C++ han sido llevados a Linux muchos otros lenguajes de programación tanto interpretados como compilados, como Smalltalk, FORTRAN, Pascal, LISP, Scheme, Ada (si es tan masoquista como para programar en Ada, no le detendremos). Además hay disponibles varios ensambladores para escribir código del 80386 en modo protegido. Los lenguajes favoritos de UNIX como Perl (el lenguaje de guiones para terminar con todos los lenguajes de guiones) y Tcl/Tk (un procesador de órdenes al estilo del intérprete de comandos incluyendo soporte para desarrollar aplicaciones simples para X Window).

El depurador avanzado `gdb` también ha sido llevado a Linux. Permite examinar los programas para localizar errores, o examinar la causa de un "cuelgue" usando un volcado del sistema (core dump). `gprof` utilidad de perfilado que permite obtener estadísticas de prestaciones de sus programas, permitiendo saber en que parte de su programa se va la mayor parte del tiempo de ejecución. El editor Emacs proporciona un entorno interactivo de edición y compilación para varios lenguajes de programación. Otras herramientas incluyen GNU `make` e `imake`, usados para dirigir el proceso de compilación de aplicaciones grandes ó RCS, sistema de control de revisiones.

Linux dispone de librerías compartidas enlazadas dinámicamente, lo que permite a los ejecutables ser mucho menores al enlazar el código de las librerías en tiempo de ejecución. Estas librerías DLL (Dinamically Linked Library) también permiten al programador de aplicaciones sustituir funciones ya definidas con su propio código. Por ejemplo, si un programador desea escribir su propia versión de la función de librería `malloc()`, el editor de enlaces usaría la nueva rutina del programador en lugar de la que se encuentra en las librerías.

Linux es ideal para desarrollar aplicaciones UNIX, proporciona un moderno entorno de programación con todos los detalles y funcionalidad necesarios. Se soportan varios estandards como POSIX.1, permitiendo a los programas escritos para Linux ser fácilmente llevados a otros sistemas. Los programadores profesionales de UNIX y administradores de sistemas pueden usar Linux para desarrollar programas en casa y luego transferir los programas a los sistemas UNIX del trabajo. Esto no solo puede ahorrar una gran cantidad de tiempo y dinero, sino que también le permitirá trabajar con la comodidad de su propia casa.[5] Los estudiantes de informática pueden usar Linux para aprender la programación en UNIX y explorar otros aspectos del sistema como la arquitectura del núcleo.

Con Linux, no solo tienes acceso a un conjunto completo de librerías y utilidades de programación, también tiene acceso al código fuente completo del núcleo y las librerías.

1.4.4 El sistema X Window

El sistema X Window es un interface gráfico estandard para máquinas UNIX. Es un potente entorno que soporta muchas aplicaciones. Usando X Windows, el usuario puede tener múltiples terminales a la vez en ventanas sobre la pantalla, cada una conteniendo una sesión diferente. Un dispositivo apuntador como el ratón es usado a menudo en el interface X, pero no es indispensable.

Han sido escritas muchas aplicaciones especificamente para X, como juegos, utilidades gráficas, herramientas de programación y documentación, y muchas más. Con Linux y X, su sistema es una auténtica estación de trabajo. Junto con una red TCP/IP puede incluso visualizar aplicaciones que se están ejecutando en otras máquinas en su pantalla local, tal y como es posible con otros sistemas corriendo X.

El sistema X Window fué desarrollado originalmente en el MIT y es de libre distribución. A pesar de esto muchas empresas han distribuido sus mejoras particulares al diseño original de X Window. La versión de X Window disponible para Linux es conocida como XFree86, una adaptación de X11R5[6] de libre distribución para sistemas UNIX basados en 80386, como es Linux. XFree86 soporta una gran variedad de tarjetas de video, incluyendo VGA, Super VGA y gran cantidad de tarjetas aceleradoras de video. Esta es una distribución completa de X Windows conteniendo el servidor de X, muchas aplicaciones y utilidades, librerías de programación y documentación.

Aplicaciones X estandard incluyen `xterm` (emulador de terminal usado por la mayoría de las aplicaciones en modo texto dentro de X); `xdm` (el gestor de sesiones, maneja los 'login'); `xclock` (un reloj simple); `xman` (un lector de páginas de manual para X) y muchos más. El número de aplicaciones disponibles para X Windows en Linux es demasiado numeroso como para mencionarlas aquí, pero la distribución básica de XFree86 incluye las aplicaciones 'estandard' que se encontraban en la versión original del MIT. Muchas otras están disponibles separadamente y teóricamente cualquier aplicación escrita para X Windows debería compilar limpiamente bajo Linux.

El aspecto y comportamiento del interface de las X Windows está controlado por el **controlador de ventanas**. Este amigable programa es el encargado de la situación de las ventanas, redimen-

[5] El autor usa su sistema Linux para desarrollar y probar aplicaciones X Window en casa, que pueden ser directamente compiladas en estaciones de trabajo en cualquier parte.

[6] N. del T.: Linux dispone en la actualidad de la versión X11R6 de X Window

sionado, iconización, movimiento de las ventanas, aspecto de los marcos de las ventanas, etc. La distribución estandard XFree86 incluye `twm`, el clásico controlador de ventanas del MIT, aunque hay disponibles controladores mucho más avanzados como Open Look Virtual Window Manager (`olvwm`). Un controlador de ventanas muy popular entre los usuarios de Linux es el `fvwm`. Es un pequeño controlador que requiere menos de la mitad de la memoria usada por `twm`. Proporciona aspecto de 3 D para las ventanas, así como un escritorio virtual —si el usuario desplaza el ratón al borde de la pantalla, la pantalla entera es desplazada como fuese mucho más grande de lo que realmente es. `fvwm` es altamente configurable y permite acceso a todas las funciones tanto desde el teclado como desde el ratón. Muchas distribuciones de Linux usan `fvwm` como controlador de ventanas estandard.

La distribución XFree86 contiene librerías para programación y ficheros de cabecera para aquellos programadores mañosos que deseen desarrollar aplicaciones para X. Están soportados varios conjuntos de controles como los de Athena, Open Look y Xaw3D. Todas las fuentes estandard, mapas de bits, páginas de manual y documentación está incluida. PEX (interface de programación para gráficos 3 D) también está soportado.

Muchos programadores de aplicaciones para X usan el conjunto comercial de controles Motif para el desarrollo. Algunos vendedores proporcionan licencias simples o multi-usuario de una versión ejecutable de Motif para Linux, pero como Motif en si, es bastante caro, no lo poseen demasiados usuarios de Linux. A pesar de todo, ejecutables estáticamente enlazados con las librerías de Motif pueden ser libremente distribuidos. Por lo tanto, si escribes un programa usando Motif y deseas distribuirlo libremente, deberás de proporcionar un ejecutable con las librerías enlazadas estáticamente para que los usuarios que no posean Motif puedan usar el programa.

La única advertencia con las X Windows son los requerimientos de máquina y memoria. Un 386 com 4 Mb de RAM es capaz de ejecutar las X, pero para usarlo adecuadamente serán necesarios 8 Mb o más. También es recomendable usar un procesador rápido, pero tener mayor cantidad de memoria física es mucho más importante. Además, para conseguir realmente buenas prestaciones de video, es muy recomendable usar una tarjeta aceleradora (como las tarjetas local bus con aceleradores gráficos S3). Se han conseguido índices de prestaciones por encima de los 140.000 Xstones con XFree86 y Linux. Con suficiente hardware, encontrará que ejecutar X en Linux es tan rápido o más rápido que ejecutar X en otras estaciones de trabajo UNIX.

En el Capítulo 5 se tratará sobre como instalar y usar las X en su propio sistema.

1.4.5 Redes

¿Está interesado en comunicarse con el mundo? ¿Si? ¿No? ¿Quizá? Linux dispone de los dos principales protocolos de red para sistemas UNIX: **TCP/IP** y **UUCP**. TCP/IP (para los aficionados a los acrónimos, Transmission Control Protocol/Internet Protocol) es un conjunto de protocolos de red que permite a sistemas de todo el mundo comunicarse en una única red conocida como Internet. Con Linux, TCP/IP y una conexión a la red, puede comunicarse con usuarios y máquinas por toda Internet mediante correo electrónico, noticias (USENET news), transferencias de ficheros con FTP y mucho más. Actualmente hay muchos sistemas Linux conectados a Internet.

La mayoría de las redes TCP/IP usan Ethernet como tipo de red física de transporte. Linux da soporte a muchas tarjetas de red Ethernet e interfaces para ordenadores personales, incluyendo el adaptador Ethernet D-Link de bolsillo para ordenadores portátiles.

Pero dado que no todo el mundo tiene una conexión Ethernet en casa, Linux también proporciona **SLIP** [7] (Serial Line Internet Protocol), el cual permite conectarse a Internet a través de un modem. Para poder usar SLIP, necesitará tener acceso a un servidor de SLIP, una máquina conectada a la red que permite acceso de entrada por teléfono. Muchas empresas y universidades tienen servidores SLIP disponibles. De hecho, si su sistema Linux dispone de conexión Ethernet y de modem, puede configurarlo como servidor de SLIP para otros usuarios.

NFS (Network File System) permite fácilmente compartir ficheros con otras máquinas de la red. FTP (File Transfer Protocol) permite la transferencia de ficheros entre máquinas. Otras aplicaciones incluyen `sendmail`, sistema para enviar y recibir correo electrónico usando el protocolo SMTP; Sistemas de noticias (news) basados en NNTP como C-News e INN; `telnet`, `rlogin` y `rsh`, que permiten iniciar una sesión y ejecutar órdenes en otras máquinas de la red; y `finger`, que permite obtener información de otros usuarios de Internet. Hay literalmente cientos de aplicaciones basadas en TCP/IP y protocolos disponibles por la red.

Todos los lectores de correo y noticias están disponibles para Linux, como `elm`, `pine`, `rn`, `nn` y `tin`. Sea cual sea tus preferencias, puedes configurar tu sistema Linux para enviar y recibir correo electrónico y noticias por todo el mundo.

Si tienes experiencia con aplicaciones TCP/IP en otros sistemas UNIX, Linux te será muy familiar. El sistema proporciona el interface estandar de programación por 'sockets', lo que virtualmente permite que cualquier programa que use TCP/IP pueda ser llevado a Linux. El servidor Linux de X también soporta TCP/IP, permitiendo ver aplicaciones que están corriendo en otros sistemas sobre tu pantalla.

En el Capítulo 5 trataremos sobre la configuración y puesta en marcha de TCP/IP, incluyendo SLIP para Linux.

UUCP (UNIX-to-UNIX Copy) es un viejo mecanismo usado para transferir ficheros, correo electrónico y noticias entre máquinas UNIX. Clásicamente las máquinas UUCP conectan entre ellas mediante líneas telefónicas y modem, pero UUCP es capaz de funcionar también sobre una red TCP/IP. Si no tienes acceso a una red TCP/IP o a un servidor SLIP, puedes configurar tu sistema para enviar y recibir ficheros y correo electrónico usando UUCP. Ver el Capítulo 5 para más información.

1.4.6 Programas de comunicaciones y BBS

Si tienes un modem serás capaz de comunicarte con otras máquinas usando alguno de los paquetes de comunicaciones disponibles para Linux. Mucha gente usa programas de comunicaciones para acceder a BBSs, así como a servicios comerciales en línea, como Prodigy, CompuServe y America On-Line. Otras personas usan sus modems para conectar a un sistema UNIX en el trabajo o escuela. Puedes usar incluso tu modem y sistema Linux para enviar y recibir FAX. Los programas de comunicaciones para Linux son muy parecidos a los que se pueden encontrar para MS-DOS y

[7] N. del T.: Linux también dispone, entre otros, de soporte PPP: Point to Point Protocol.

otros sistemas operativos. Todo aquel que haya usado alguna vez un paquete de comunicaciones encontrará un equivalente familiar en Linux.

Uno de los paquetes de comunicaciones más populares para Linux es `Seyon`, una aplicación para X que proporciona un interface configurable y ergonómico con soporte incorporado de varios protocolos de transferencia de ficheros como Kermit, ZModem y otros. Otros programas de comunicaciones incluyen C-Kermit, `pcomm` y `minicom`. Estos últimos son similares a programas de comunicaciones que se pueden encontrar en otros sistemas operativos y son muy fáciles de usar.

Si no tienes acceso a un servidor de SLIP (ver la sección anterior), puedes usar `term` para multiplexar la línea serie. `term` te permitirá abrir múltiples sesiones a través de la conexión por modem a una máquina remota. `term` también te permitirá redireccionar las conexiones de un cliente X a tu servidor local de X, a través de la línea serie, permitiéndote visualizar aplicaciones X remotas en tu terminal local. Otro paquete disponible, KA9Q, proporciona un interface similar al SLIP.

Mantener un sistema de BBS es el hobby favorito de mucha gente. Linux proporciona una gran variedad de programas de BBS, la mayoría de los cuales son más potentes que los disponibles para otros sistemas operativos. Con una línea telefónica, un modem y Linux, puedes convertir tu sistema en una BBS, proporcionando acceso telefónico a tu sistema a usuarios de todo el mundo. Los programas de BBS para Linux incluyen XBBS y UniBoard BBS.

La mayoría de los programas de BBS mantienen al usuario en un sistema basado en menús donde solo hay disponibles ciertas funciones y programas. Una alternativa a este acceso a la BBS es el acceso UNIX completo, lo que significa que el usuario llama y entra en el sistema como un usuario regular (a través de un "login"). Aunque este procedimiento requiere mayor mantenimiento y atención por parte del administrador del sistema, puede hacerse, y proporcionar acceso público a UNIX desde el sistema Linux no es difícil. Junto con una red TCP/IP, puedes proporcionar acceso a correo electrónico y noticias a los usuarios del sistema.

Si no tienes acceso a una red TCP/IP o UUCP, Linux también te permitirá comunicarte con la gran cantidad de redes de BBS, como FidoNet, en la cual puedes intercambiar correo electrónico y noticias por la línea telefónica. En el Capítulo 5 encontrarás más información sobre programas para comunicaciones y BBS bajo Linux.

1.4.7 Enlazando con MS-DOS

Existen varias utilidades para enlazar con el mundo de MS-DOS. El más conocido es el emulador de MS-DOS para Linux, que permite ejecutar muchas aplicaciones de MS-DOS directamente desde Linux. Aunque Linux y MS-DOS son dos sistemas operativos totalmente diferentes, el modo protegido del 80386 permite que ciertas tareas se comporten como si estuviesen en un 8086 tal y como hacen las aplicaciones de MS-DOS.

El emulador de MS-DOS está todavía bajo desarrollo, aunque muchos de los programas más populares corren sobre él. Por lo tanto, es comprensible que programas MS-DOS que usan características extrañas o esotéricas del sistema, puede que nunca estén soportados, puesto que solo es un emulador. Por ejemplo, no espere que sea posible ejecutar programas que usan las características del 80386 en modo protegido como Microsoft Windows (en modo mejorado de 386).[8]

Aplicaciones que corren sin problemas bajo el emulador de MS-DOS incluyen 4DOS (intérprete de comandos), Foxpro 2.0, Harvard Graphics, MathCad, Stacker 3.1, Turbo Assembler, Turbo C/C++, Turbo Pascal, Microsoft Windows 3.0 (en modo *real*) y WordPerfect 5.1. Los comandos y utilidades estandard de MS-DOS (como `PKZIP`,.. etc) funcionan también con el emulador.

El emulador de MS-DOS es principalmente una solución 'ad-hoc' para aquellos usuarios que necesitan MS-DOS solo para unas pocas aplicaciones, pero usan Linux para todo lo demás. Eso no significa que no sea una implementación completa de MS-DOS. Por supuesto, si el emulador no satisface tus necesidades, siempre se puede usar MS-DOS junto con Linux en el mismo sistema. Usando el gestor de arranque LILO puedes especificar en el arranque que sistema operativo se desea arrancar. Linux puede coexistir con otros sistemas operativos, como OS/2, etc.

Linux proporciona un interface sin fisuras para transferir ficheros entre Linux y MS-DOS. Se puede montar una partición MS-DOS o disco flexible bajo Linux y acceder directamente a los ficheros MS-DOS como se haría con ficheros de Linux.

Actualmente existe en desarrollo un proyecto conocido como **WINE**— que es un emulador de Microsoft Windows para X Windows bajo Linux. Cuando WINE esté completado, los usuarios podrán ejecutar aplicaciones MS-Windows directamente desde Linux, similar al emulador de Windows WABI de Sun Microsystems. Al tiempo de escribir este texto, WINE está todavía en las primeras etapas del desarrollo, pero tiene muy buenas perspectivas.

En el Capítulo 5 hablaremos de las herramientas MS-DOS disponibles para Linux.

1.4.8 Otras aplicaciones

Tal y como se podría esperar de tal mezcolanza de sistema operativo, hay una gran cantidad de miscelánea disponible para Linux. La intención primaria de Linux es, actualmente, el uso personal de UNIX, pero esto está cambiando rápidamente. Programas de negocios y científicos se están expandiendo y los vendedores de programas comerciales están comenzando a contribuir al crecimiento del número de aplicaciones disponibles para Linux.

Están disponibles para Linux varias bases de datos relacionales, incluyendo Postgres, Ingres y Mbase. Estas tienen todas las características de aplicaciones de bases de datos cliente/servidor similares a las que se pueden encontrar en otras plataformas UNIX. `/rdb`, un sistema de base de datos comercial[9] está también disponible.

Aplicaciones de cálculo científico incluyen FELT (herramienta de análisis de elementos finitos); `gnuplot` (aplicación de análisis y representación gráfica de datos); Octave (matemática simbólica, similar a MATLAB); `xspread` (hoja de cálculo); `fractint`, versión para X del popular generador de fractales Fractint; `xlispstat` (paquete estadístico), y muchos más. Otras aplicaciones incluyen a Spice (herramienta de diseño y análisis de circuitos) y Khoros (proceso de señales imágenes/digital

[8]N. del T.: Determinadas aplicaciones Windows 3.1 funcionan, aunque en fase experimental, bajo diversos emuladores

[9]N. del T.: En el momento de realizar esta traducción ya existían varios paquetes de Bases de Datos comerciales portadas a Linux. Van desde compatibilidad con Clipper 5.x hasta soporte SQL entre otros

y sistema de visualización).

Por supuesto, hay muchas más aplicaciones que han sido y pueden ser llevadas a Linux. Sea cual sea su campo de trabajo, transladar aplicaciones UNIX a Linux debería ser bastante directo. Linux proporciona un completo entorno de programación UNIX, suficiente para servir de base a aplicaciones científicas.

Como con cualquier sistema operativo, Linux tiene su porción de juegos. Estos incluyen juegos clásicos en modo texto como Nethack y Moria; MUDs (dragones y mazmorras multiusuario, que permiten que los usuarios interaccionen en una aventura en modo texto) como DikuMUD y Tiny-MUD; así como juegos para X Windows como `tetris`, `netrek`, y `Xboard` (la versión para X11 de `gnuchess`). El popular juego 'arcade' de masacrar enemigos *Doom* también ha sido llevado a Linux.

Para audíofilos, Linux tiene soporte para tarjetas de sonido y programas relacionados como CD-player (programa que permite escuchar CD musicales en el lector de CD-ROM), secuenciadores MIDI y editores (permiten componer música para sintetizadores o cualquier instrumento con controlador MIDI) y editores de sonidos digitalizados.

¿No puedes encontrar el programa que estás buscando?. El 'Linux Software Map' descrito en el Apéndice A contiene una lista con gran cantidad de programas que han sido escritos o llevados a Linux. Ciertamente, la lista está lejos de ser completa, pero contiene una gran cantidad de programas. Otra forma de encontrar aplicaciones para Linux, si tiene acceso a Internet, es buscar en los ficheros `INDEX` que se encuentran en los 'Linux FTP sites'. A poco que hurgue a su alrededor, encontrará gran cantidad de programas esperando a ser usados.

Si de ninguna forma puede encontrar lo que necesita, siempre puede tratar de hacer una adaptación de la aplicación desde otra plataforma a Linux. La mayoría de los programas de libre distribución para UNIX compilarán en Linux sin ningún problema. O si todo falla, puede escribirse Vd. mismo el programa. Si lo que está buscando es un programa comercial, puede que exista un 'clon' de libre difusión disponible. O puede sugerir a la empresa que considere el lanzamiento de una versión de su programa para Linux. Algunas personas han contactado con empresas de programación pidiéndoles que lleven sus aplicaciones a Linux, y han obtenido varios grados de éxito.

1.5 Acerca del Copyright

Linux está regido por lo que se conoce como la *Licencia Pública General* de GNU, o *GPL, General Public License.* La GPL fue desarrollada para el proyecto GNU por la *Free Software Foundation*, que podemos traducir como "Fundación por el Software Gratuito". La licencia hace una serie de previsiones sobre la distribución y modificación del "software gratis". "Gratis" en este sentido se refiere a libertad, y no solo coste. La GPL puede ser interpretada de distintas formas, y esperamos que este resumen le ayude a entenderla y cómo afecta a Linux. Se incluye una copia completa de la Licencia al final del libro, en el apéndice F.

Originalmente, Linus Torvalds lanzó Linux bajo una licencia más restrictiva que la GPL, que permitía que el software fuera libremente distribuido y modificado, pero prohibía su uso para ganar dinero. Sin embargo, la GPL autoriza que la gente venda su software, aunque no le permite restringir el derecho que su comprador tiene a copiarlo y venderlo a su vez.

En primer lugar, hay que aclarar que el "software gratis" de la GPL *no es* software de dominio público. El software de dominio público carece de *copyright* y pertenece literalmente al público. El software regido por la GPL sí tiene el copyright de su autor o autores. Esto significa que está protegido por las leyes internacionales del copyright y que el autor del software está declarado legalmente. No solo porque un programa sea de libre distribución puede considerársele del dominio público.

El software regido por la GPL tampoco es "shareware". Por lo general, el "shareware" es propiedad del autor, y exige a los usuarios que le paguen cierta cantidad por utilizarlo después de la distribución. Sin embargo, el software que se rige por la GPL puede ser distribuido y usado sin pagar a nadie.

La GPL permite a los usuarios modificar el software y redistribuirlo. Sin embargo, cualquier trabajo derivado de un programa GPL se regirá también por la GPL. En otras palabras, una compañía nunca puede tomar Linux, modificarlo y venderlo bajo una licencia restringida. Si un software se deriva de Linux, éste deberá regirse por la GPL también.

La GPL permite distribuir y usar el software sin cargo alguno. Sin embargo, también permite que una persona u organización gane dinero distribuyendo el software. Sin embargo, cuando se venden programas GPL, el distribuidor no puede poner ninguna restricción a la redistribución. Esto es, si usted compra un programa GPL, puede a su vez redistribuirlo gratis o cobrando una cantidad.

Esto puede parecer contradictorio. ¿Por qué vender software cuando la GPL especifica que puede obtenerse gratis? Por ejemplo, supongamos que una empresa decide reunir una gran cantidad de programas GPL en un CD-ROM y venderlo. La empresa necesitará cobrar por el hecho de haber producido el CD, y asímismo querrá ganar dinero. Esto está permitido por la GPL.

Las organizaciones que vendan el software regido por la GPL deben tener en cuenta algunas restricciones. En primer luger, no pueden restringir ningún derecho al comprador del programa. Esto significa que si usted compra un CD-ROM con programas GPL, podrá copiar ese CD y revenderlo sin ninguna restricción. En segundo lugar, los distribuidores deben hacer saber que el software se rige por la GPL. En tercer lugar, el vendedor debe proporcionar, sin coste adicional, el código fuente del software a distribuir. Esto permite a cualquiera comprar el software y modificarlo a placer.

Permitir a una empresa distribuir y vender programas que son gratis es bueno. No todo el mundo tiene acceso a Internet para llevarse los programas, como Linux, gratis. La GPL permite a las empresas vender y distribuir programas a esas personas que no pueden acceder al software con un coste bajo. Por ejemplo, muchas empresas venden Linux en disquetes o CD-ROM por correo, y hacen negocio de esas ventas. Los desarrolladores de Linux pueden no tener constancia de estos negocios. Por ejemplo, Linus sabe que ciertas compañías venden Linux, y él no va a cobrar nada por esas ventas.

En el mundo de los programas gratis, lo importante no es el dinero. El objetivo es permitir desarrollar y distribuir software fantástico asequible a cualquiera. En la siguiente sección, hablaremos de cómo esto se aplica al desarrollo de Linux.

1.6 Diseño y filosofía de Linux

En ocasiones, los nuevos usuarios de Linux se crean falsas expectativas acerca de éste. Linux es un sistema operativo único, y es importante entender su filosofía y diseño para usarlo eficientemente. Aunque usted sea un experimentado "gurú" del UNIX, lo que viene a continuación le interesará con total seguridad.

En versiones comerciales de UNIX, el sistema es desarrollado siguiendo una rigurosa política de mantenimiento de la calidad, con sistemas de control de revisiones para las fuentes y documentación, etc. Los desarrolladores no pueden añadir cosas nuevas por su cuenta: cualquier cambio será en respuesta a un informe de *bug* detectado, y será cuidadosamente registrado en el sistema de control de versiones, de manera que podrá volverse atrás sin problemas. Cada desarrollador tiene asignada una o más partes del código, y solo ese desarrollador puede alterar esas secciones del código.

Internamente, el departamento de calidad realiza conjuntos de tests para cada versión del sistema operativo, e informa de los errores. Es responsabilidad de los desarrolladores corregir esos errores. Se utiliza un complicado sistema de análisis estadístico para asegurarse de que cierto porcentaje de los errores son corregidos antes de la próxima versión.

Como vemos, el proceso seguido por los desarrolladores de UNIX comercial para mantenerlo y darle soporte es muy complicado, pero razonable. La compañía debe tener cierta seguridad de que la próxima revisión del sistema ya está lista para ser comercializada, a través de las pruebas que hemos comentado. Esto supone un gran trabajo que involucra a cientos (si no miles) de programadores, *betatesters*, redactores de documentación y personal administrativo. Por supuesto, no todos los fabricantes de UNIX trabajan así, pero esto da una idea de la panorámica habitual.

Con Linux, hay que olvidarse del concepto de desarrollo organizado, sistemas de control de versiones, información de errores estructurada o análisis estadísticos. Linux es un sistema operativo hecho por "*hackers*"[10]

Linux ha sido desarrollado principalmente por un grupo de programadores de todo el mundo unidos por Internet. A través de Internet, cualquiera tiene la oportunidad de unirse al grupo y ayudar al desarrollo y depuración del núcleo, portar nuevo software, escribir documentación o ayudar a los nuevos usuarios. La comunidad de Linux se comunica principalmente mediante diversas listas de correo y grupos de USENET. Existen algunos acuerdos en el desarrollo, como que cualquiera que desee que su código sea incluido en la versión "oficial" del núcleo deberá ponerse en contacto con Linus Torvalds, quien comprobará el código y lo incluirá en el núcleo. Por lo general, estará encantado en hacerlo, siempre que no estropee otras cosas.

El sistema se ha diseñado siguiendo una filosofía abierta y de crecimiento. Por regla general hay una nueva versión del núcleo cada dos meses (a veces cada menos tiempo)[11] Esto depende del número de errores a corregir, la cantidad de información recibida de los usuarios, y lo que haya dormido Linus esta semana.

Por lo tanto, es difícil decir que todos los errores hayan sido corregidos en cierta fecha. Pero

[10] Por "hacker" queremos referirnos a programadores fervientemente dedicados, por *hobby*, a explotar sus ordenadores al máximo, con resultados útiles para otras personas. Este concepto es contrario al habitualmente aceptado, que dice que un "hacker" es un pirata informático.

[11] N. del T.: Actualmente nos consta la aparición semanal de al menos una revisión de las versiones *experimentales*.

conforme el sistema va apareciendo libre de errores críticos o manifiestos, se considera "estable" y se lanzan nuevas revisiones. Hay que recordar que no se pretende realizar un sistema perfecto, sin errores. Simplemente, se pretende desarrollar una implementación gratuita de UNIX. Linux está hecho *para* desarrolladores, más que para otro tipo de personas.

Si alguien desarrolla una aplicación o nueva característica para el núcleo, se añade inicialmente en una etapa "alfa", es decir, pensada ser probada por usuarios atrevidos que aceptan los fallos que tenga. Dado que la comunidad de Linux se basa sobre todo en Internet, el software "alfa" es enviado, normalmente, a servidores FTP dedicados a Linux (vea el apéndice C) y anunciado mediante un mensaje puesto en un grupo USENET dedicado a Linux. Los usuarios que se llevan el software lo prueban e informan sobre errores o sugerencias al autor por correo electrónico.

Una vez que se corrigen los problemas iniciales, el código pasa a una etapa "beta", en la que se considera estable pero incompleto (o sea, funciona, pero no incluye todas las funcionalidades previstas). También se puede pasar a una etapa "final" en la que el software se considera terminado. Para el código del núcleo, una vez que se ha terminado, el desarrollador puede pedir a Linus que lo incluya en el núcleo estándar, o bien que lo ponga como código optativo.

Recordar que lo anterior son solo convenciones, y no leyes. Algunos programas pueden no necesitar fases "alpha". Es el desarrollador quien tomará las decisiones al respecto.

Estará sorprendido de que un grupo de programadores y aficionados voluntarios, relativamente desorganizados, hayan podido hacer algo. Sin embargo, este grupo constituye uno de los más eficientes y motivados. Todo el núcleo de Linux ha sido escrito *de la nada*, sin emplear ningún código de fuentes propietarias. Todo el software, librerías, sistemas de ficheros y drivers han sido desarrollados o portados desde otros sistemas; y se han programado drivers para los dispositivos más populares.

Normalmente, Linux es distribuido junto con otro software en lo que se conoce como *distribución*, que es un paquete de software que permite poner a punto un sistema completo. Dado que es difícil para los usuarios noveles el crear un sistema UNIX partiendo del núcleo y programas de diversas fuentes, se crearon las distribuciones con el fin de facilitar esta tarea: con la distribución, usted solo tiene que coger el CD-ROM o los disquetes e instalar para tener un sistema completo con programas de aplicación incluidos. Como es de esperar, no hay ninguna distribución "estándar". Hay muchas, cada una con sus ventajas y desventajas. Hablaremos más sobre distribuciones en la sección 2.1.

A pesar de lo anterior, necesitará cierto conocimiento de UNIX para poder instalar y usar un sistema completo. No hay distribución que no tenga fallos, por lo que requerirá de su habilidad al teclado para corregir los pequeños problemas que se presenten. Ejecutar un sistema UNIX no es fácil, ni siquiera en las versiones comerciales. Hay que andarse con cuidado en cualquier UNIX, y Linux no es una excepción.

1.6.1 Consejos para novatos en UNIX

Instalar y utilizar su propio sistema Linux no requiere grandes conocimientos de UNIX. De hecho, muchos novatos de UNIX instalan con éxito sus sistemas. Es una experiencia de la que se aprende, pero de la que se pueden generar fustraciones. Si tiene suerte, podrá instalar y usar Linux sin conocimiento alguno de UNIX. Sin embargo, cuando quiera introducirse en tareas más complejas (instalar nuevo software, recompilar el núcleo, etc.) el conocimiento de UNIX empieza a ser necesario.

Por suerte, ejecutando su propio sistema Linux podrá aprender las cosas esenciales de UNIX para llevar a cabo esas tareas. Este libro contiene información que intentará ayudarle (vea el capítulo 3, que es un tutorial sobre UNIX; y el capítulo 4, que le enseñará cosas sobre la administración de Linux). Quizás le parezca interesante leer esos capítulos antes de instalar Linux. En este caso, se ahorrará algunos problemas.

Nadie puede esperar hacerse experto administrador de la noche a la mañana. No hay implementación alguna que esté libre de mantenimiento. Por lo tanto, prepárese si es novato, pero no se desespere ante los problemas. El que escribe esto[12] tampoco tenía mucha idea antes de empezar.

1.6.2 Consejos para los *gurús*

La gente con años de experiencia programando y administrando sistemas UNIX también puede necesitar ayuda al querer instalar Linux. Existen aspectos con los que los magos del UNIX tendrán que familiarizarse antes de entrar en materia. Por un lado, Linux no es un sistema UNIX comercial. No intenta seguir exactamente el mismo estándar de otros sistemas. En otras palabras, este factor no es el *único* factor que interesa en el desarrollo de Linux.

Es más importante, sin embargo, su funcionalidad. En muchos casos, el nuevo código introducido en el núcleo aun tendrá errores o estará incompleto. Se acepta que es más importante lanzar el código cuanto antes para que los usuarios lo puedan probar. Por ejemplo, el emulador WINE (emulador de Microsoft Windows para Linux) tiene una versión alfa "oficial" mientras la prueban los usuarios. En este sentido, la comunidad de Linux tiene derecho a trabajar con este código, probarlo y ayudar a su desarrollo. Los UNIX comerciales, raramente siguen esta política.

Si ha sido administrador de UNIX durante más de diez años, y ha usado diferentes sistemas comerciales, verá que este sistema es muy moderno y dinámico. Se crea un nuevo núcleo cada dos meses aproximadamente. Se lanza nuevo software permanentemente. Un día puede creer que su sistema está a la última, y al día siguiente creer que está en la Edad de Piedra.

Con toda esta actividad dinámica, ¿cómo puede mantenerse al día? La mayor parte de las veces, es mejor actualizarse el software de manera incremental; o sea, actualizar aquellas partes del sistema que *necesita* actualizar. Por ejemplo, si nunca va a usar Emacs, no hay motivo para que se actualice continuamente ese programa. Incluso siendo usuario de Emacs, no hay necesidad de actualizarlo cada vez, a menos que haya alguna característica nueva que le interese. No hay razón para estar siempre cambiándose las versiones del software.

Esperamos que Linux satisfaga sus expectativas de un sistema UNIX. El corazón de Linux alberga el espíritu del software gratuito y el desarrollo constante. La comunidad de Linux favorece su expansión, y es difícil creer en su fiabilidad, especialmente cuando se procede del UNIX comercial. Tampoco puede esperarse que Linux sea perfecto, ya que no lo es ningún otro software. Sin embargo, creemos que Linux es tan útil y completo como cualquier otra implementación de UNIX.

[12] N. del T.: al igual que el que lo traduce

1.7 Diferencias Entre Linux y Otros Sistemas Operativos

Es importante entender las diferencias entre Linux y otros sistemas operativos, tales como MS-DOS, OS/2, y otras implementaciones de UNIX para ordenador personal. Antes de nada, conviene aclarar que Linux puede convivir felizmente con otros sistemas operativos en la misma máquina: es decir, Ud. puede correr MS-DOS y OS/2 en compañía de Linux sobre el mismo sistema sin problemas. Hay incluso formas de interactuar entre los diversos sistemas operativos como veremos.

1.7.1 ¿Por qué usar Linux?

¿Por qué usar Linux en lugar de un sistema operativo comercial conocido, bien probado, y bien documentado? Podríamos darle miles de razones. Una de las más importantes es, sin embargo, que Linux es una excelente elección para trabajar con UNIX a nivel personal. Si Ud. es un desarrollador de software UNIX, ¿por qué usar MS-DOS en casa? Linux le permitirá desarrollar y probar el software UNIX en su PC, incluyendo aplicaciones de bases de datos y X Window. Si es Ud. estudiante, la oportunidad está en que los sistemas de su universidad correrán UNIX. Con Linux, podrá correr su propio sistema UNIX y adaptarlo a sus necesidades. La instalación y uso de Linux es también una excelente manera de aprender UNIX si no tiene acceso a otras máquinas UNIX.

Pero no perdamos la vista. Linux no es solo para los usuarios personales de UNIX. Es robusto y suficientemente completo para manejar grandes tareas, asi como necesidades de cómputo distribuidas. Muchos negocios—especialmente los pequeños—se están cambiando a Linux en lugar de otros entornos de estación de trabajo basados en UNIX. Las universidades encuentran a Linux perfecto para dar cursos de diseño de sistemas operativos. Grandes vendedores de software comercial se están dando cuenta de las oportunidades que puede brindar un sistema operativo gratuito.

Las siguientes secciones deberían destacar las diferencias más importantes entre Linux y otros sistemas operativos. Esperamos que encuentre que Linux puede cubrir sus necesidades, o (al menos) mejorar su entorno actual de trabajo. Tenga en mente que la mejor manera de saborear Linux es simplemente probandolo—ni siquiera necesita instalar un sistema completo para tener una muestra de él. En el Capítulo 2, le mostramos como.

1.7.2 Linux vs. MS-DOS

No es raro tener ambos, Linux y MS-DOS, en el mismo sistema. Muchos usuarios de Linux confian en MS-DOS para aplicaciones tales como procesadores de texto. Aunque Linux proporciona sus propios análogos para estas aplicaciones (por ejemplo, TEX), existen varias razones por las que un usuario concreto desearía correr tanto MS-DOS como Linux. Si toda su tesis está escrita en WordPerfect[13] para MS-DOS, puede no ser capaz de convertirla fácilmente a TEX o algún otro formato. Hay muchas aplicaciones comerciales para MS-DOS que no están disponibles para Linux, y no hay ninguna razón por la que no pueda usar ambos.

Como puede saber, MS-DOS no utiliza completamente la funcionalidad de los procesadores 80386 y 80486. Por otro lado, Linux corre completamente en el modo protegido del procesador y explota

[13] N. del T.: El ejemplo del autor se ha quedado obsoleto: Ya existe un WordPerfect 6.1 nativo para Linux

todas las características del éste. Puede acceder directamente a toda su memoria disponible (e incluso más allá de la disponible, usando RAM virtual). Linux proporciona un interface UNIX completo no disponible bajo MS-DOS—el desarrollo y adaptación de aplicaciones UNIX bajo Linux es cosa fácil, mientras que, bajo MS-DOS, Ud. está limitado a un pequeño subgrupo de la funcionalidad de programación UNIX. Al ser Linux un verdadero sistema UNIX, Ud. no tendrá estas limitaciones.

Podríamos debatir los pros y contras de MS-DOS y Linux durante páginas y páginas. Sin embargo, baste decir que Linux y MS-DOS son entidades completamente diferentes. MS-DOS no es caro (comparado con otros sistemas operativos comerciales), y tiene un fuerte asentamiento en el mundo de los PC's. Ningún otro sistema operativo para PC ha conseguido el nivel de popularidad de MS-DOS—básicamente porque el coste de esos otros sistemas operativos es inaccesible para la mayoría de los usuarios de PC's. Muy pocos usuarios de PC pueden imaginar gastarse 100000 ptas. o más solamente en el sistema operativo. Linux, sin embargo, es gratis, y por fin tiene la oportunidad de decidirse.

Le permitiremos emitir sus propios juicios de Linux y MS-DOS basados en sus expectativas y necesidades. Linux no está destinado a todo el mundo. Si siempre ha querido tener un sistema UNIX completo en casa, si es alto el coste de otras implementaciones UNIX para PC, Linux puede ser lo que estaba buscando.

Se dispone de herramientas que permiten interactuar entre Linux y MS-DOS. Por ejemplo, es fácil acceder a los ficheros MS-DOS desde Linux. También hay disponible un emulador de MS-DOS, que permite ejecutar muchas aplicaciones populares de MS-DOS. Actualmente se está desarrollando un emulador de Microsoft Windows.

1.7.3 Linux vs. Otros Sistemas Operativos

Están surgiendo un gran número de sistemas operativos avanzados en el mundo del PC. Concretamente, OS/2 de IBM y Windows NT de Microsoft comienzan a tener popularidad a medida que los usuarios de MS-DOS migran a ellos.

Ambos, OS/2 y Windows NT son sistemas operativos completamente multitarea, muy parecidos a Linux. Técnicamente, OS/2, Windows NT y Linux son bastante similares: Soportan aproximadamente las mismas características en términos de interfaz de usuario, redes, seguridad, y demás. Sin embargo, la diferencia real entre Linux y los otros es el hecho de que Linux es una versión de UNIX, y por ello se beneficia de las contribuciones de la comunidad UNIX en pleno.

¿Qué hace a UNIX tan importante? No solo es el sistema operativo más popular para máquinas multiusuario, también es la base de la mayoría del mundo del software de libre distribución. Si tiene acceso a Internet, casi todo el software de libre distribución disponible está específicamente escrito para sistemas UNIX. (Internet en si está profundamente basada en UNIX.)

Hay muchas implementaciones de UNIX, de muchos vendedores, y ni una sola organización es responsable de su distribución. Hay un gran pulso en la comunidad UNIX por la estandarización en forma de sistemas abiertos, pero ninguna corporación controla este diseño. Por eso, ningún vendedor (o, como parece, ningún hacker) puede adoptarr estos estándares en una implementación de UNIX.

Por otro lado, OS/2 y Windows NT son sistemas propietarios. El interface y diseño están controlados por una sola corporación, y solo esa coproración puede implementar ese diseño. (No espere encontrar una versión gratis de OS/2 en un futuro cercano.) De alguna forma, este tipo de organización es beneficiosa: establece un estándar estricto para la programación y el interfaz de usuario distinto al encontrado incluso en la comunidad de sistemas abiertos. OS/2 es OS/2 vaya donde vaya—lo mismo ocurre con Windows NT.

Sin embargo, el interface UNIX está constantemente desarrollandose y cambiando. Varias organizaciones están intentando estandarizar el modelo de programación, pero la tarea es muy difícil. Linux, en particular, es en su mayoría compatible con el estandar POSIX.1 para el interface de programación UNIX. A medida que pase el tiempo, se espera que el sistema se adhiera a otros estandars, pero la estandarización no es la etapa primaria en la comunidad de desarrollo de Linux.

1.7.4 Otras implementaciones de UNIX

Hay otras implementaciones de UNIX para el 80386 y 80486. La arquitectura 80386 se presta al diseño de UNIX y buen un número de vendedores han sacado ventaja de este factor.

Hablando de características, otras implementaciones de UNIX para PC son bastante similares a Linux. Podrá ver que casi todas las versiones comerciales de UNIX soportan basicamente el mismo software, entorno de programación, y características de red. Sin embargo, hay algunas fuertes diferencias entre Linux y las versiones comerciales de UNIX.

En primer lugar, Linux soporta un rango de hardware diferente de las implementaciones comerciales. En general, Linux soporta la mayoría de dispositivos hardware conocidos, pero el soporte está aun limitado a ese hardware al que los desarrolladores tengan acceso actualmente. Sin embargo, los vendedores de UNIX comercial por lo general tienen una base de soporte más amplia, y tienden a soportar más hardware, aunque Linux no esta tan lejos de ellos. Cubriremos los requerimientos hardware de Linux en la Sección 1.8.

En segundo, las implementaciones comerciales de UNIX suelen venir acompañadas de un conjunto completo de documentación, así como soporte de usuario por parte del vendedor. Por contra, la mayoría de documentación para Linux se limita a documentos disponibles en Internet—y libros como este. En la Sección 1.9 se listan fuentes de documentación de Linux y otra información.

En lo que concierne a estabilidad y robustez, muchos usuarios han comentado que Linux es al menos tan estable como los sistemas UNIX comerciales. Linux está aun en desarrollo, y algunas características (como redes TCP/IP) son menos estables, pero mejora a medida que pasa el tiempo.

El factor más importante a considerar por muchos usuarios es el precio. El software de Linux es gratis, si tiene accesso a Internet (o a otra red de ordenadores) y puede traerselo. Si no tiene acceso a tales redes, tiene la opción de comprarlo pidiéndolo por correo en disquetes, cinta o CD-ROM (vea el Apéndice B). Por supuesto, Ud. puede copiarse Linux de un amigo que puede tener ya el software, o compartir el coste de comprarlo con alguien más. Si planea instalar Linux en un gran número de máquinas, solo necesita comprar una copia del software—Linux no se distribuye con licencia para "una sola máquina".

El valor de las implementaciones comerciales de UNIX no debería ser rebajado: conjuntamente con el precio del software en sí, uno paga generalmente por la documentación, el soporte, y una etiqueta de calidad. Estos factores son muy importantes para grandes instituciones, pero los usuarios de ordenadores personales pueden no necesitar esos beneficios. En cualquier caso, muchos negocios y universidades encuentran que correr Linux en un laboratorio con ordenadores baratos es preferible a correr una versión comercial de UNIX en un laboratorio de estaciones de trabajo. Linux es capaz de proporcionar la funcionalidad de una estación de trabajo sobre hardware de PC a una fracción de su coste.

Como un ejemplo del "mundo real" sobre el uso de Linux dentro de la comunidad informática, los sistemas Linux han viajado hasta los grandes mares del Pacífico Norte, encargandose de las telecomunicaciones y análisis de datos para en un buque de investigación oceanográfica. Los sistemas Linux se están usando en estaciones de investigación en la Antártida. Como ejemplo más mundano, quizá, varios hospitales están usando Linux para mantener registros de pacientes. Está probando ser tan fiable y útil como otras implementaciones de UNIX.

Hay otras implementaciones gratuitas o baratas de UNIX para el 386 y 486. Una de las más conocidas es 386BSD, una implementación y adaptación del UNIX BSD para el 386. 386BSD es comparable a Linux en muchos aspectos, pero cual de ellos es "mejor" depende de las necesidades y espectativas personales. La única distinción fuerte que se puede hacer es que Linux se desarrolla abiertamente (donde cualquier voluntario puede colaborar en el proceso de desarrollo), mientras 386BSD se desarrolla dentro de un equipo cerrado de programadores que mantienen el sistema. Debido a esto, existen diferencias filosóficas y de diseño serias entre los dos proyectos. Los objetivos de los dos proyectos son completamente distintos: el objetivo de Linux es desarrollar un sistema UNIX completo desde el desconocimiento (y divertirse mucho en el proceso), y el objetivo de 386BSD es en parte modificar el código de BSD existente para usarlo en el 386.

NetBSD es otra adaptación de la distribución NET/2 de BSD a un número de máquinas, incluyendo el 386. NetBSD tiene una estructura de desarrollo ligeramente más abierta, y es comparable al 386BSD en muchos aspectos.

Otro proyecto conocido es HURD, un esfuerzo de la Free Software Foundation (Fundación de Software gratuito) para desarrollar y distribuir una versión gratis de UNIX para muchas plataformas. Contacte con la Free Software Foundation (la dirección se da en el Apéndice F) para obtener más información sobre este proyecto. Al tiempo de escribir este documento, HURD aun está en los primeros pasos de su desarrollo.

También existen otras versiones baratas de UNIX, tales como Coherent (disponible por cerca de $99) y Minix (un clónico de UNIX académico, pero útil, en el que se basaron los primeros pasos del desarrollo de Linux). Algunas de estas implementaciones son de interés en mayor parte académico, mientras otras son sistemas ya maduros para productividad real. No hace falta decir, sin embargo, que muchos usuarios de UNIX de PC se están pasando a Linux.

1.8 Requerimientos de Hardware

Ahora debe estar convencido de lo maravilloso que es Linux y de las estupendas cosas que puede hacer por usted. Sin embargo, antes de que se precipite e instale el software, tiene que asegurarse de los requerimientos y limitaciones de hardware de Linux.

Tenga en cuenta que Linux fue desarrollado por sus usuarios. Esto siginifica, en la mayoría de las ocasiones, que el hardware soportado por Linux es únicamente aquél al que usuarios y desarrolladores tienen realmente acceso. Según van apareciendo, se van soportando la mayor parte del hardware y los periféricos populares para sistemas 80386/80486 (de hecho, Linux soporta más hardware que algunas implementaciones comerciales de UNIX). Sin embargo, aún no son reconocidos algunos de los periféricos más oscuros y esotéricos. Según pasa el tiempo aumenta el hardware soportado por Linux, de manera que si sus periféricos favoritos no están aquí listados, hay gran probabilidad de que sean soportados próximamente.

Otro inconveniente en el soporte de hardware bajo Linux es que muchas compañías han decidido conservar las especificaciones del interfaz de su hardware como propietario. Como consecuencia de ésto, los desarrolladores voluntarios de Linux simplemente no pueden escribir controladores (drivers) para esos periféricos (y si pudieran, tales controladores serían propiedad de la compañía dueña del interfaz, lo cual violaría el GPL). Las compañías que mantienen interfaces propietarios, escriben sus propios controladores para sistemas operativos como MS-DOS y Microsoft Windows; el usuario final (es decir, usted) no necesita saber nada del interfaz. Desafortunadamente, esto impide a los desarrolladores de Linux escribir controladores para esos periféricos.

Hay muy poco que hacer en esta tesitura. En algunos casos, los programadores han intentado escribir controladores "cloneados" basándose en asumciones acerca del interfaz. En otras ocasiones, los desarrolladores trabajarán con la compañía en cuestión e intentarán obtener información del interfaz del periférico con un nivel de éxito que puede variar.

En las siguientes secciones, trataremos de resumir los requerimientos hardware para Linux. El documento *Hardware HOWTO* (ver Sección 1.9) contiene un listado más completo del hardware soportado por Linux.

Aclaración: un buen tratado acerca del hardware soportado por Linux, está actualmente en fase de desarrollo. Algunas distribuciones puede que soporten, o no, estas características experimentales. Esta sección lista en primer lugar el hardware que ya lleva cierto tiempo soportado y del que se sabe de su estabilidad. En caso de duda, consulte la documentación de la distribución de Linux que está usando (ver Sección 2.1 para más información acerca de las distribuciones de Linux).

1.8.1 Requisitos de Placa Base y de CPU

Actualmente Linux soporta sistemas con una CPU Intel[14] 80386, 80486, o Pentium. Esto incluye todas las variantes del tipo de CPU, como el 386SX, 486SX, 486DX, y 486DX2. Los "clónicos" no Intel, como AMD y Cyrix también funcionan con Linux.

Si tiene un 80386 o 80486SX, puede que también quiera aprovechar el coprocesador matemático, si bien no es imprescindible (el núcleo de Linux puede emular el coprocesador si no cuenta con uno). Están soportados todos los copros estándar, tales como los IIT, Cyrix FasMath, e Intel.

La placa base debe ser de arquitectura ISA o EISA en cuanto a bus se refiere. Estos términos definen cómo interactúa el sistema con los periféricos y otros componentes por medio del bus principal. La mayoría de los sistemas vendidos hoy son de bus ISA o EISA. El bus MicroChannel (MCA),

[14] N. del T.: Linux ya soporta, en la actualidad a otras arquitecturas tales como ALPHA, Amiga, PowerPc, etc.

que se encuentra en máquinas como los IBM/PS2 no está soportado actualmente.

Los sistemas que usan arquitectura de bus local (para accesos más rápidos en vídeo y disco) también están soportados. Es recomendable que tenga una arquitectura de bus local estándar como pueda ser el Bus Local VESA[15](“VLB”).

1.8.2 Requisitos de memoria

Linux, comparado con otros sistemas operativos avanzados, necesita muy poca memoria para funcionar . Debería contar con un mínimo de 2 megabytes de RAM; sin embargo, es altamente recomendable tener 4 megabytes. Cuanta más memoria tenga más rápido irá su sistema.

Linux soporta el rango completo de direcciones de 32-bits de los 386/486; es decir, utilizará toda la memoria RAM de forma automática, y se apañará sin problemas con tan sólo 4 megabytes de RAM, incluyendo aplicaciones “pedigüeñas” como X-Window, Emacs, y demás. Sin embargo, disponer de más memoria es casi tan importante como tener un procesador más rápido. 8 megabytes es más que suficiente para uso personal; 16 megabytes o más pueden ser necesarios si espera una fuerte carga de usuarios en el sistema.

La mayoría de los usuarios de Linux, reservan una parte del disco duro para espacio de intercambio (“swapping”) que se usa como RAM virtual. Incluso si dispone de bastante memoria RAM física en su máquina, puede que quiera utilizar un área de “swap” . El área de “swap” no puede reemplazar a una memoria física RAM real, pero puede permitir a su sistema ejecutar aplicaciones más grandes guardando en disco duro aquellas partes de código que están inactivas. La cantidad de espacio de intercambio a reservar depende de diversos factores; volveremos a esta cuestión en la Sección 2.2.3.

1.8.3 Requisitos de la controladora de disco duro

No necesita un disco duro para ejecutar Linux; puede ejecutar un sistema mínimo completamente desde disquete. Sin embargo, resulta lento y muy limitado, y de todas formas, muchos usuarios tienen acceso a almacenamiento en disco duro. Debe tener una controladora AT-estándar (16-bit). El núcleo soporta controladoras XT-estándar (8 bit); aunque la mayoría de las controladoras usadas hoy son AT-estándar. Linux debería soportar todas las controladoras MFM, RLL, e IDE. La mayoría, pero no todas, las controladoras ESDI están soportadas—sólo aquellas que hacen emulación hardware de la ST506.

La regla general para controladoras que no sean SCSI, es que si puede acceder a las unidades (disco y/o disquete) desde MS-DOS u otro sistema operativo, debería poder hacerlo desde Linux.

Linux también soporta un número de controladoras de disco SCSI, si bien el soporte para SCSI es más limitado a causa de la gran cantidad de estandars que existen para el interfaz de las controladoras. Las controladoras SCSI soportadas[16] incluyen las Adaptec AHA1542B, AHA1542C, AHA1742A (versión de BIOS 1.34), AHA1522, AHA1740, AHA1740 (controladora SCSI-2, BIOS

[15]N. del T.: PCI también está soportado

[16]N. del T.: Ver la sección 1.9 sobre como conseguir el fichero *Hardware HOWTO* que contendrá una lista más actualizada y completa

1.34 en modo mejorado); Future Domain 1680, TMC-850, TMC-950; Seagate ST-02; UltraStor SCSI; Western Digital WD7000FASST. Las controladoras clónicas basadas en estas tarjetas también deberían funcionar.

1.8.4 Requisitos de espacio en disco

Efectivamente, para instalar Linux, necesitará tener algo de espacio libre en su disco duro. Linux soporta múltiples discos duros en la misma máquina; puede disponer de espacio para Linux en múltiples unidades si es necesario.

La *cantidad* de espacio en disco duro que necesitará depende en gran medida de sus necesidades y de la cantidad de software que va a instalar. Linux es relativamente pequeño en relación a las implementaciones de UNIX; usted podría correr un sistema completo con 10-20 megabytes de espacio en disco. Sin embargo, si quiere disponer de espacio para expansiones, y para paquetes más grandes como X Window, necesitará más espacio. Si planea permitir a múltiples usuarios utilizar la máquina, tendrá que dejar espacio para sus ficheros.

También, a menos que tenga un montón de memoria RAM física (16 megabytes o más), necesitará crear espacio de intercambio ("swap"), para ser usado como RAM virtual. Comentaremos todos estos detalles acerca de la instalación y uso del espacio de intercambio en la Sección 2.2.3.

Cada distribución de Linux normalmente viene con algún que otro texto que debería ayudarle a estimar la cantidad precisa de espacio a reservar en función del software que planee instalar. Puede ejecutar un sistema mínimo con menos de 20 megabytes; un sistema completo con toda la parafernalia en 80 megabytes o menos; y un sistema grande con sitio para muchos usuarios y espacio para futuras expansiones en un rango de 100-150 megabytes. De nuevo, estas cifras son meramente orientativas; usted tendrá que decidir, según sus necesidades y objetivos, los requerimientos específicos de almacenamiento para su sistema.

1.8.5 Requisitos de monitor y adaptador de vídeo

Linux soporta todas las tarjetas de vídeo estándar Hercules, CGA, EGA, VGA, IBM monocromo, y Super VGA así como monitores para el interfaz por defecto basado en texto. En general, si la combinación que tiene de monitor y tarjeta de vídeo funcionan bajo otro sistema operativo como MS-DOS, debería funcionar perfectamente con Linux. La genuínas tarjetas CGA de IBM sufren el (d)efecto nieve ("snow") bajo Linux, por lo que no es muy recomendable su uso.

Los entornos gráficos como el Sistema X Window tienen requerimientos propios de hardware para la tarjeta de vídeo. En vez de listar estos requerimientos aquí, los trataremos en la Sección 5.1.1. De forma breve, para ejecutar el Sistema X Window en su máquina Linux, necesitará alguna de las tarjetas de vídeo listadas en esa sección.

1.8.6 Hardware diverso

Las secciones de arriba describían qué hardware es necesario para correr un sistema Linux. Sin embargo, la mayoría de los usuarios tienen cierto número de dispositivos “opcionales” como almacenamiento en cintas y CD-ROM, tarjetas de sonido y demás, y están interesados en saber si este hardware está, o no es, soportado por Linux. Siga leyendo.

1.8.6.1 Ratones y otros dispositivos apuntadores

Normalmente usted sólo usará un ratón bajo un entorno gráfico como el Sistema X Window. Sin embargo, algunas aplicaciones Linux no asociadas con un entorno gráfico, hacen uso del ratón.

Linux soporta todos los ratones serie estándar, incluyendo Logitech, MM series, Mouseman, Microsoft (2 botones) y Mouse Systems (3 botones). Linux también soporta los ratones de bus Microsoft, Logitech, y ATIXL. El interfaz de ratón de PS/2 también es soportado.

Todos los demás dispositivos apuntadores, como los “trackball” que emulen a los dispositivos de arriba, también deberían funcionar.

1.8.6.2 Almacenamiento en CD-ROM

Casi todas las unidades de CD-ROM usan el interfaz SCSI[17]. Siempre y cuando tenga un adaptador SCSI soportado por Linux, su unidad de CD-ROM debería funcionar. Hay un número de unidades de CD-ROM que se ha comprobado funcionan bajo Linux, incluyendo el NEC CDR-74, Sony CDU-541, y Texel DM-3024. Las unidades internas Sony CDU-31a y Mistsumi también están soportadas por Linux.

Linux soporta el sistema de ficheros estándar para CD-ROMs ISO-9660.

1.8.6.3 Unidades de cinta

Hay varios tipos de unidades de cinta disponibles en el mercado. La mayoría usan el interfaz SCSI, por lo que estarían soportadas por Linux. Entre las unidades de cinta SCSI verificadas están la Sankyo CP150SE; Tandberg 3600; Wangtek 5525ES, 5150ES, y 5099EN con el adaptador PC36. Otras unidades QIC-02 deberían estar soportadas también.

Existen controladores en fase de desarrollo para varios otros dispositivos de cinta como las unidades Colorado que se conectan a la controladora de disquetes.

1.8.6.4 Impresoras

Linux soporta todas las impresoras paralelo. Si puede acceder a su impresora por el puerto paralelo desde MS-DOS, u otro sistema operativo, debería poder acceder a él desde Linux también. El

[17] N. del T.: El parque de CD ROMs ha cambiado de forma asombrosa en los últimos meses. Sirva decir que Linux soporta, además de las citadas en el párrafo, los nuevos estándares ATAPI para CD ROMs conectables a controladoras IDE

software de impresión de Linux consiste en el software estándar de UNIX `lp` y `lpr`. Este software también le permite imprimir remotamente a través de la red, si es que tiene una disponible.

1.8.6.5 Modems

Igual que para las impresoras, Linux soporta toda la gama de modems serie, tanto internos como externos. Hay una gran cantidad de software de telecomunicaciones disponible para Linux, incluyendo Kermit, `pcomm`, `minicom`, Seyon. Si su modem es accesible desde otro sistema operativo en la misma máquina, debería poder acceder a él sin dificultad desde Linux.

1.8.7 Tarjetas Ethernet

Linux soporta un buen número de tarjetas Ethernet y adaptadores para LAN[18]. Esto incluye los siguientes:

- 3com 3c503, 3c503/16
- Novell NE1000, NE2000
- Western Digital WD8003, WD8013
- Hewlett Packard HP27245, HP27247, HP27250
- D-Link DE-600

Los siguientes clónicos se ha informado que funcionan:

- LANNET LEC-45
- Alta Combo
- Artisoft LANtastic AE-2
- Asante Etherpak 2001/2003,
- D-Link Ethernet II
- LTC E-NET/16 P/N 8300-200-002
- Network Solutions HE-203,
- SVEC 4 Dimension Ethernet
- 4-Dimension FD0490 EtherBoard 16

Aquellas tarjetas clónicas compatibles con cualquiera de las tarjetas anteriores también debería funcionar.

[18] N. del T.: Algunas tarjetas token-ring ya están soportadas

1.9 Fuentes de información sobre Linux

Como podrá imaginar, además de este libro, hay muchas otras fuentes de información sobre Linux. Concretamente, hay numerosos libros sobre UNIX en general, que recomendamos a aquellos lectores que no tengan experiencia previa con UNIX. Si somos nuevos en UNIX, lo más indicado es leer uno de estos libros antes de meternos en la "peligrosa selva" de Linux. Un buen comienzo puede ser el libro *Learning the UNIX Operating System*, de Grace Todino y John Strang.

Casi todas las fuentes de información sobre Linux están disponibles principalmente de forma electrónica. Esto es, deberá tener acceso a una red, como Internet, USENET o Fidonet, con el fin de obtener la documentación. Si no tiene acceso a ninguna red, siempre puede encontrar la forma de obtener copias impresas en disquetes o CDROM de los libros.

1.9.1 Documentación *On-Line*

Si tiene acceso a Internet, encontrará variada documentación en muchos servidores de FTP del mundo. Si no tiene acceso directo a Internet, aun puede obtener los documentos: muchos distribuidores de Linux en CDROM incluyen toda o casi toda la documentación existente en la red. Además, se suelen distribuir por redes diferentes como Fidonet o Compuserve. Y si tiene acceso únicamente al correo en Internet, puede obtener ficheros de servidores FTP sin más que usar un servidor de `ftpmail`. Vea el apéndice C para más información.

Hay gran cantidad de servidores FTP que distribuyen software y documentación de Linux. En el apéndice C encontrará una lista con servidores conocidos. Con el fin de reducir el tráfico de red, debería utilizar el servidor que le quede más cercano[19] geográficamente.

El apéndice A incluye una lista de algunos de los documentos sobre Linux que se encuentran disponibles por FTP anónimo. Los nombres de los ficheros pueden no ser los mismos en todos los servidores, pero suelen estar en el directorio `docs` dentro del directorio que dediquen a Linux. Por ejemplo, en `sunsite.unc.edu` los ficheros de Linux están en `/pub/Linux` y la documentación en `/pub/Linux/docs`.

Algunos documentos que puede encontrar son las *Linux FAQ*, una colección de FAQ sobre Linux; los documentos *HOWTO*, dedicados a aspectos específicos, como la instalación inicial (*Installation HOWTO*) o la configuración de impresoras y spoolers (*Printing HOWTO*).

Algunos documentos se envían regularmente a uno o más grupos USENET sobre Linux. No deje de leer la sección 1.9.4 sobre el tema de las News.

1.9.2 Linux en el WWW

La página inicial de la documentación de Linux en el Web se encuentra en la dirección URL

`http://sunsite.unc.edu/mdw/linux.html`

[19] N. del T.: Vea el apéndice E.1 para localizar una lista de ftps españoles

Desde esta página puede accederse a los *HOWTO*s y otros documentos en formato HTML. También se encuentran enlaces a otros servidores de interés.

1.9.3 Libros y otras publicaciones

En este momento, hay algunos trabajos publicados sobre Linux. Principalmente, los libros del Proyecto de Documentación de Linux (LDP), que se lleva a cabo mediante Internet para escribir y distribuir una colección de manuales para Linux. Estos manuales son análogos a los que se publican junto con versiones comerciales de UNIX: tratan la instalación y puesta en marcha, programación, trabajo en red, asuntos del núcleo y muchas cosas más.

Los manuales del LDP se encuentran disponibles mediante FTP anónimo en el Internet, así como por correo a través de algunos comercios. En el apéndice A se enumeran los manuales disponibles y cómo conseguirlos.

No hay muchos más libros que traten el tema particular de Linux. Sin embargo, sí que hay numerosos libros sobre UNIX en general que normalmente son aplicables a Linux, como aquellos sobre cómo utilizar o programar sobre el sistema UNIX, ya que Linux no difiere mucho en su interfaz con el usuario o programador. En resumen, lo que quiera saber sobre el uso y programación de Linux lo encontrará en los libros sobre UNIX. Encontrará los nombres de algunos de estos libros en el apéndice A.

Con los libros escritos para Linux y los que hay sobre UNIX se debe estar ya suficientemente documentado como para hacer cualquier cosa sobre su sistema Linux.

También existe un *magazine* mensual sobre Linux, el *Linux Journal*. Se distribuye por todo el mundo y es una excelente manera de mantenerse al día en este tema, sobre todo si no se tiene acceso a USENET. En el apéndice A encontrará información sobre cómo suscribirse a esta publicación.

1.9.4 Grupos de NEWS USENET

"USENET" es un foro mundial de artículos electrónicos organizado en "grupos", o sea, áreas de discusiones relacionadas con cada tema concreto. Buena parte del desarrollo de Linux ha sido a través de Internet y USENET, con lo que no es extraño que existan bastantes grupos que traten el tema.

Inicialmente, el grupo sobre Linux era `alt.os.linux`, y se creó para tratar aquí las cuestiones que sobre Linux abundaban ya en `comp.os.minix` y varias listas de correo. El tráfico en el grupo de Linux fue creciendo lo suficiente como para permitirse el paso a la jerarquía `comp`, en Febrero de 1992.

`comp.os.linux` se ha convertido en un grupo de News muy conocido, más que cualquiera otro de `comp.os`. En Diciembre del 92 se votó la creación del grupo `comp.os.linux.announce` para reducir el tráfico de `comp.os.linux`. En Julio de 1993 se partió este grupo de forma definitiva en la jerarquía que hoy existe.

Si no tiene acceso a USENET, pero sí puede usar el correo electrónico, existen pasarelas de correo a News disponibles para cada uno de los grupos siguientes.

`comp.os.linux.announce`
: `comp.os.linux.announce` es un grupo moderado, pensado para anuncios importantes respecto a Linux (como informes sobre errores detectados, lanzamiento de parches, etc). Si quiere leer grupos de Linux, empiece por éste. Los artículos que aquí se publican no son reenviados a ningún otro grupo normalmente. En él se pueden encontrar además muchos artículos que se envían periódicamente, incluyendo documentos ya mencionados como los *HOWTO*s.

  Los envíos al grupo deben ser aceptados por los moderadores, Matt Welsh y Lars Wirzenius. Si quiere enviar algo, normalmente basta con que lo ordene a su software de News. Este software se ocupará de enviar el artículo a los moderadores para que lo acepten. Sin embargo, si su sistema no está correctamente configurado, puede enviarlo directamente a la dirección de correo `linux-announce@tc.cornell.edu`.

  Los restantes grupos sobre Linux no están moderados.

`comp.os.linux.help`
: Es el grupo más popular. Contiene preguntas y respuestas sobre la instalación o ejecución de los sistemas Linux. Si tiene problemas con Linux, puede usar este grupo y recibirá respuesta de quien pueda ayudarle. Sin embargo, se recomienda que lea antes la documentación por si ahí encuentra la ayuda necesaria.

`comp.os.linux.admin`
: En este grupo se discuten temas relacionados con la administración del sistema Linux como entorno multiusuario. Aquí son bienvenidos los comentarios y preguntas sobre software de empaquetado, de backups, o sobre la gestión de cuentas de usuarios.

`comp.os.linux.development`
: En este grupo se discuten asuntos relacionados con el desarrollo en el núcleo de Linux. Por ejemplo, si está escribiendo un driver nuevo y necesita consejo sobre algún aspecto, es aquí donde deberá preguntar. Además, es aquí donde se discuten los objetivos en el desarrollo de Linux, tal y como se dice en la sección 1.6.

  Debe observarse que este grupo no está pensado para tratar el desarrollo de software *para* Linux, sino el desarrollo *del propio* Linux. Estos temas deben tratarse en otro grupo.

`comp.os.linux.misc`
: Aquí se mantienen debates que no encajen en los grupos anteriores. Concretamente, las típicas guerras "Linux vs. Windows 95" o cualquier artículo de carácter no técnico.

El grupo `comp.os.linux` sigue existiendo, pero se recomienda usar los nuevos grupos. Si no tiene acceso a ellos, hable con su administrador de las News.

1.9.5 Listas de correo en Internet

Si tiene acceso al correo electrónico de Internet, puede aun participar en las listas de correo aunque no tenga acceso a USENET. A estas listas de correo puede apuntarse incluso sin tener acceso alguno a Internet, gracias a las pasarelas que ofrecen otros servicios, como UUCP, FidoNET o CompuServe.

La lista de correo de "Activistas del Linux" [20] está pensada principalmente para desarrolladores y gente interesada en participar en el desarrollo. Es una lista "multicanal", en la que puede apuntarse a uno o más "canales" según los temas que le interesen. Entre estos canales se encuentran el `NORMAL`, el `KERNEL`, el `GCC` (sobre el compilador de GNU), el `NET` o el `DOC` (sobre el tema de la documentación de Linux).

Para más información sobre esta lista de correo, envíe un mensaje a

```
linux-activists@niksula.hut.fi
```

Recibirá una lista de canales disponibles e información de cómo apuntarse o borrarse de éstos.

Existen otras listas de correo de propósito especial. El mejor camino para encontrarlas es leer los anuncios aparecidos en USENET, y la lista de "mailing-list" disponible periódicamente en el grupo `news.answers`.

1.10 Cómo obtener ayuda

Indudablemente, necesitará cierta ayuda durante sus primeras aventuras en el mundo de Linux. Aquí veremos algunas indicaciones sobre cómo obtener esa ayuda.

La forma más inmediata de buscar ayuda es mediante las listas de correo y grupos de USENET que se mantienen en Internet, tal como explicamos en la sección 1.9. Si no tiene acceso a Internet, puede encontrar ayuda en otros servicios on-line como los BBS's o Compuserve.

También hay ciertas empresas que han empezado a dar soporte comercial de Linux. Esto le permitirá pedir ayuda a los técnicos a cambio del pago de una cuota. El apéndice B contiene una lista de comerciantes de Linux, alguno de los cuales ofrece soporte técnico. Sin embargo, si tiene acceso a USENET y al correo de Internet, verá que puede obtener soporte técnico de calidad y gratuito.

Lo que sigue son sugerencias que le hacemos para mejorar su experiencia con Linux y garantizarle mayor probabilidad de éxito para encontrar soluciones a los problemas que se le presenten.

Consultar primero toda la documentación disponible

Lo primero que debe hacer cuando encuentre un problema es leerse la documentación que se lista en la sección 1.9 y el apéndice A. Estos documentos fueron laboriosamente escritos para gente como usted—gente que necesitaba ayuda sobre Linux. Los libros sobre Unix también se pueden usar para resolver dudas de Linux.

Frecuentemente, y por difícil que parezca, se encuentran respuestas a muchos problemas.

[20] N. del T.: Linux Activists

Si se tiene acceso a las *news* de USENET o a listas de correo relacionadas con Linux, se debe *leer* su contenido antes de poner una pregunta sobre sus problemas. Muchas veces hay problemas comunes que no se encuentran fácilmente en los libros pero que tienen frecuente respuesta en los grupos de USENET o similar. Enviar la pregunta a los grupos sin leerlos antes puede ser una pérdida de tiempo.

Aprenda a apreciar las cosas hechas por uno mismo.

En la mayoría de los casos se recomienda investigar por cuenta de uno mismo antes de pedir ayuda al exterior. Recuerde que Linux no es un sistema comercial y puede hacer con él lo que quiera (modificarlo, etc). Si aprende a hacerse las cosas por sí mismo, su experiencia le permitirá llegar a ser, tal vez, uno de los famosos "gurús" de Linux.

Mantenga la calma.

Hay que evitar la desesperación a pesar de todos los problemas. No se ha oído aun ninguna historia de alguien que, en un ataque de ira, borrase todos sus discos de Linux con un fuerte imán. Los mismos autores se han desahogado a puñetazos con almohadas o similares en arrebatos de este tipo. Hay que esperar un poco a que las distribuciones de Linux se hagan más fáciles aun de instalar, aunque ya lo son más que casi cualquier otro Unix comercial.

Evite poner preguntas en USENET demasiado pronto.

Mucha gente comete el error de pedir ayuda en USENET demasiado pronto. Cuando encuentre algún problema, no se vaya de inmediato al terminal más cercano (insistimos, *no lo haga*) para escribir su duda en un grupo de News de Linux. Es mejor que intente antes resolverlo pues muchas veces es debido al nerviosismo inicial y se puede caer en el error de preguntar cosas demasiado triviales. Vamos, que si su PC no se enciende, mire antes a ver si está enchufado.

Si escribe su pregunta en USENET, haga que merezca la pena.

Finalmente, puede que se vea obligado a pedir ayuda a los foros dedicados a Linux, bien mediante listas de correo o con USENET. Cuando redacte su pregunta, recuerde que la gente que va a leerla no está ahí para ayudarle. La Red no es un servicio post-venta. Así pues, hay que redactar el artículo de la forma más completa posible.

¿Cómo puede conseguir esto? En primer lugar, debe incluir toda la información acerca de su sistema y el problema que crea relevante. Una escueta pregunta como "no puedo conseguir que el correo electrónico funcione" difícilmente será contestada, a menos que incluya información acerca de su sistema, qué software utiliza, qué ha intentado hacer para resolverlo y qué resultados ha obtenido. Además, suele ser interesante añadir información más general, como qué versión del sistema utiliza (del núcleo y distribución), así como un pequeño resumen de su hardware. Pero tampoco exagere incluyendo su tipo de monitor, por ejemplo, cuando su problema sea con el software de red.

En segundo lugar, recuerde que necesita intentar resolver por su cuenta el problema antes de recurrir a la Red. Si, por ejemplo, nunca ha intentado instalarse el correo electrónico, y pretende que directamente le cuenten en la Red cómo se hace, estará cometiendo un gran error. Hay muchos documentos disponibles (vea la sección 1.9) acerca de cómo iniciar la instalación de paquetes de aplicación típicos bajo Linux. La idea es hacer todo lo que pueda por sí mismo y *solo entonces* pedir ayuda si finalmente la necesita.

Recuerde también que la gente que vea su mensaje, generalmente dispuesta a ayudarle, puede hartarse de ver la misma pregunta una y otra vez. Observe las preguntas y respuestas que existen en el grupo de News actualmente, pues muchas veces la solución a su problema ya está en la Red antes de que lo pregunte.

Por último sea conciso, en la medida de lo posible, a la hora de redactar un artículo. Es mucho más efectivo ser breve, directo e informativo al pedir ayuda. No olvide la educación. Son frecuentes las disputas en la Red sin venir demasiado a cuento. La Red[21] es un excelente medio para buscar ayuda y para resolver sus problemas de Linux pero hay que saber utilizarla.

[21] N. del T.: La Red se refiere a Internet, como la red que engloba a todas las redes públicas de datos

Capítulo 2

Obtención e Instalación de Linux

En este capítulo describiremos cómo obtener el software de Linux, en forma de una de las distribuciones ya preparadas, y cómo instalar la distribución que elija.

Como hemos citado, no hay una única distribución "oficial" del software de Linux; hay, de hecho, muchas distribuciones, cada una de las cuales sirve a un propósito particular y una serie de objetivos. Estas distribuciones están disponibles por FTP anónimo en Internet, en BBS de todo el mundo, y por correo en disquete, cinta, y CD-ROM.

Aquí presentamos una visión general del proceso de instalación. Cada distribución tiene sus propias instrucciones de instalación específicas, pero armado con los conceptos presentados aquí debería ser capaz de sentirse cómodo frente a cualquier instalación. El Apéndice A lista las fuentes de información de instrucciones de instalación y otra ayuda, por si Ud. está completamente perdido.

Este libro contiene secciones adicionales detallando la distribución Slackware de Linux.

2.1 Distribuciones de Linux

Al ser Linux un software de libre distribución, no hay ninguna organización o entidad responsable de mantenerlo y distribuirlo. Por tanto, cualquiera es libre de agrupar y distribuir el software, en tanto en cuanto respete las restricciones de la GPL. El resultado final de ésto es que existen muchas distribuciones de Linux, disponibles a través de FTP anónimo o pidiéndolo por correo.

Ud. se encuentra ahora con la tarea de decidirse por una distribución en particular de Linux que se ajuste a sus necesidades. No todas las distribuciones son iguales. Muchas de ellas incluyen prácticamente todo el software que Ud. necesitaría para poner en marcha un sistema completo— y algunas otras distribuciones son "pequeñas" distribuciones orientadas a usuarios sin copiosas cantidades de espacio en disco. Muchas distribuciones solamente contienen lo esencial del software de Linux, y se espera que Ud. instale por su propia cuenta paquetes de software más grandes, como el Sistema X-Window.

(En el Capítulo 4 le mostraremos cómo.)

El Linux *Distribution HOWTO* (vea el Apéndice A) contiene una lista de las distribuciones de Linux disponibles a través de Internet, así como por correo. El Apéndice B contiene una lista de direcciones de contacto de un número de vendedores de Linux por correo. Si Ud. compró este libro impreso, el editor debería poder proporcionarle una distribución de Linux o decirle cómo puede conseguirla.

¿Cómo puede Ud. decidirse entre todas estas distribuciones? Si tiene acceso a las news de USENET, u otro sistema de conferencias por ordenador, podría preguntar allí las opiniones personales de la gente que haya instalado Linux. Incluso mejor, si conoce a alguien que haya instalado el Linux, pídale consejo y ayuda. Hay muchos factores a considerar cuando se elije una distribución, sin embargo, las necesidades y opiniones de cada uno son diferentes. En la actualidad, la mayoría de las distribuciones populares de Linux contienen aproximadamente el mismo conjunto de software, de forma que la elección de una distribución es más o menos arbitraria.

Este libro contiene información sobre la instalación de las distribuciones populares Slackware y Slackware Pro de Linux.

2.1.1 Conseguir Linux desde Internet

Si Ud. tiene acceso a Internet, el modo más fácil de obtener Linux es a través de FTP anónimo.[1] El apéndice C contiene una lista con servidores de archivos por FTP que poseen software de Linux. Uno de éstos es `sunsite.unc.edu`, y puede encontrar varias distribuciones de Linux en el directorio:

```
/pub/Linux/distributions
```

Muchas distribuciones se crean en servidores de FTP anónimo como un conjunto de imágenes de disco. Es decir, la distribución consiste en un conjunto de ficheros, y cada fichero contiene la imagen binaria de un disquete. Para poder copiar el contenido de un fichero imagen en el disquete, se puede usar el programa `RAWRITE.EXE` bajo MS-DOS. Este programa copia, bloque a bloque, el contenido de un fichero a un disquete, sin preocuparse del formato del disco.[2]

`RAWRITE.EXE` puede encontrarlo en varios servidores de FTP de Linux, como por ejemplo en uno de los sistemas que mayor soporte brindan a Linux: `sunsite.unc.edu` en el directorio

```
/pub/Linux/system/Install/rawwrite
```

Por lo tanto, en muchos casos, simplemente puede transferirse el conjunto de imágenes de disquete, y usar `RAWRITE.EXE` con cada imagen de modo que cree un conjunto de disquettes. Arranque desde el llamado “boot diskette” y ya estará dispuesto a comenzar. El software se suele instalar directamente desde los disquetes, aunque algunas distribuciones le permiten instalarlo desde una partición MS-DOS de su disco duro. Algunas distribuciones le permiten instalar desde una red

[1] Si no tiene acceso directo a Internet, puede obtener Linux a través del servicio `ftpmail`, siempre y cuando tenga capacidad de intercambiar correo con Internet. Lea el Apéndice C para más detalle.

[2] Si tiene acceso a una estación de trabajo UNIX con una disquetera, también puede usar el comando `dd` para copiar el fichero imagen directamente al disquete. Un comando como “`dd of=/dev/rfd0 if=foo bs=18k`” hará un “raw write” de los contenidos del fichero `foo` a la disquetera en una estación de trabajo SUN. Consulte con sus administradores de UNIX para mayor información sobre las disqueteras de su sistema y el uso del comando `dd`.

TCP/IP. La documentación de cada distribución debería describir estos métodos de instalación si los tienen disponibles.

Otras distribuciones de Linux se instalan desde un conjunto de disquetes con formato MS-DOS. Por ejemplo, la distribución Slackware de Linux sólo necesita que se creen los disquetes boot y root usando `RAWRITE.EXE`. El resto de disquetes se copian a disquetes con formato MS-DOS usando el comando MS-DOS `COPY`. El sistema instala el software directamente desde los disquetes MS-DOS. Esto le ahorra el problema de tener que usar `RAWRITE.EXE` con muchos ficheros imagen, aunque, por otro lado, requiere que Ud. tenga acceso a un sistema MS-DOS para instalar el sistema.

Cada distribución de Linux disponible en FTP anónimo debería incluir un fichero `README` describiendo cómo transferir y preparar los disquetes para instalación. Asegúrese de leer toda la documentación disponible para la versión que esté usando.

Cuando transfiera el software de Linux, asegúrese de usar el modo *binary* para todas sus transferencias de ficheros (con la mayoría de clientes FTP, el comando "`binary`" activa este modo).

Lea la Sección 2.1.4, abajo, si necesita información sobre cómo obtener la distribución Slackware desde Internet.

2.1.2 Conseguir Linux desde otras fuentes online

Si tiene acceso a otra red de ordenadores como Compuserve o Prodigy, puede haber medios de transferir el software de Linux desde esas fuentes. Además, muchos sistemas BBS (Bulletin Board System) también proporcionan el software de Linux. Podrá ver una lista de BBS de Linux en el Apéndice D. No todas las distribuciones de Linux están disponibles desde estas redes de ordenadores, sin embargo— muchas de ellas, en especial las diversas distribuciones en CD-ROM, sólo pueden obtenerse a través de pedido por correo.

2.1.3 Conseguir Linux por correo

Si no tiene acceso a Internet o a una BBS, muchas distribuciones pueden conseguirse por correo en disquete, cinta, o CD-ROM. El Apéndice B lista un número de estos distribuidores. Muchos de ellos aceptan tarjetas de crédito, así como pedidos internacionales, de modo que, si no está Ud. en Estados Unidos o Canadá, aun debería poder conseguir Linux de este modo.

Linux es un software de libre distribución, aunque a los distribuidores se les permite por la GPL cargar una cantidad por ello. Por ello, pedir Linux por correo podría costarle entre 30 y 150 dólares USA, dependiendo de la distribución. Sin embargo, si Ud. conoce a alguien que ya haya comprado o se haya transferido una versión de Linux, Ud. es libre de pedírsela o copiársela para su propio uso. A los distribuidores de Linux no se les permite restringir la licencia o redistribución del software en ninguna medida. Si está pensando en instalar un laboratorio completo de máquinas Linux, por ejemplo, solo necesitará comprar una única copia de una de las distribuciones, la cual puede usar para instalar todas las máquinas.

2.1.4 Conseguir la Slackware

Slackware es una popular distribución de Linux mantenida por Patrick Volkerding.[3] Es fácil de instalar y está bastante completa, y se puede obtener a través de Internet o también en CD-ROM a través de varios vendedores (ver Apéndice B).

La distribución Slackware consta de varios "disk sets" (agrupaciones de disquetes) , cada uno de los cuales contiene un tipo particular de software (por ejemplo, el "disk set" `d` contiene herramientas de desarrollo como el compilador `gcc`, y el "disk set" `x` contiene el software del sistema X-Window). Puede elegir instalar cualquiera de los "disk sets" que prefiera e instalar otros nuevos más tarde.

La versión de Slackware descrita aquí es la 2.0.0 del 25 de Junio de 1994. La instalación de versiones posteriores de la Slackware debería ser muy similar a la que se detalla aquí.

2.1.4.1 *Sets* de discos de Slackware

Desafortunadamente, Slackware no mantiene una lista completa de los requerimientos de espacio en disco por cada uno de los conjuntos de discos. Necesitará al menos 7 megabytes para instalar sólo las series de discos "`A`"; una estimación aproximada del espacio en disco requerido sería de unos 2 ó 2.5 megabytes por disco.

Están disponibles los siguientes disk sets:

`A` El sistema base. Suficiente para instalar y echar a andar Linux teniendo el elvis y programas de comunicaciones disponibles. Basado en el Kernel 1.0.9, y en el nuevo estándar de sistema de ficheros (FSSTND).

Estos discos están preparados para caber en discos de 1.2M, aunque el resto de la Slackware no. Si sólo tiene una disquetera de 1.2M, aún puede instalar el sistema base bajándose los otros discos que desee e instalándolos desde su disco duro.

`AP` Contienen varias aplicaciones y anexos, como las páginas del manual, `groff`, `ispell` (en sus versiones internacional y GNU), `term`, `joe`, `jove`, `ghostscript`, `sc`, `bc`, y los " quota patches ".

`D` Desarrollo de programas. GCC/G++/Objective C 2.5.8, `make` (GNU y BSD), `byacc` y GNU `bison`, `flex`, las librerías 4.5.26 de C, `gdb`, fuentes del kernel 1.0.9, `SVGAlib`, `ncurses`, `clisp`, `f2c`, `p2c`, `m4`, `perl`, y `rcs`.

`E` GNU Emacs 19.25.

`F` Una colección de *FAQs* y otros documentos.

`I` Páginas de información sobre software GNU. Documentación para varios programas que pueden ser leídas con `info` o Emacs.

`N` Redes. TCP/IP, UUCP, `mailx`, `dip`, `deliver`, `elm`, `pine`, `smail`, `cnews`, `nn`, `tin`, `trn`.

[3]Se puede contactar con Patrick Volkerding por Internet en `volkerdi@mhd1.moorhead.msus.edu`.

`OOP`	Programación orientada a objetos. GNU Smalltalk 1.1.1, y el interface para Smalltalk en X (STIX).
`Q`	Fuentes de kernels Alpha e imágenes (en la actualidad contiene Linux 1.1.18).[4]
`TCL`	Tcl, Tk, TclX, blt, itcl.
`Y`	Juegos. La colección de juegos BSD, y tetris para terminales.
`X`	El sistema base XFree86 2.1.1, y además con `libXpm`, `fvwm` 1.20, y `xlock`.
`XAP`	Aplicaciones X: X11 `ghostscript`, `libgr13`, `seyon`, `workman`, `xfilemanager`, `xv` 3.01, GNU `chess` y `xboard`, `xfm` 1.2, `ghostview`, y varios juegos.
`XD`	Desarrollo de programas X11. Librerías X11, servidor linkkit y soporte para PEX.
`XV`	Xview 3.2 release 5. Librerías XView, y los gestores de ventanas virtuales y no virtuales Open Look.
`IV`	Librerías Interviews, includes, y las aplicaciones `doc` e `idraw`.
`OI`	El Object Builder 2.0 y el Object Interface Library 4.0 de ParcPlace, generosamente puestos a disposición de desarrolladores en Linux de acuerdo con los términos expresados en la nota sobre copias que se encuentra en esas librerías. Hay que tener en cuenta que esto sólo se aplica con `libc`-4.4.4, aunque puede aparecer una nueva versión una vez que la versión 2.5.9 esté disponible.
`T`	Los sistemas para el formateo de textos TEX y LATEX

Deberá conseguir el disk set "`A`"; el resto son opcionales. Nosotros sugerimos la instalación de los disk sets `A`, `AP` ,y `D`, así como el set `X` si tiene intención de utilizar el sistema X-Window.

2.1.4.2 Conseguir la Slackware por Internet

La distribución Slackware de Linux se puede encontrar en muchos FTP sites a lo largo del mundo. El Apéndice C lista algunos de los FTP sites que hay de Linux; le sugerimos que intente encontrar el software en el FTP site que esté más cercano a su zona para reducir el tráfico de la red. No obstante, dos de los mayores FTP sites sobre Linux son `sunsite.unc.edu` y `tsx-11.mit.edu`.

La distribución Slackware puede encontrarse como mínimo en los siguientes sites:

- `sunsite.unc.edu:/pub/Linux/distributions/slackware`
- `tsx-11.mit.edu:/pub/linux/packages/slackware`
- `ftp.cdrom.com:/pub/linux/slackware`

`ftp.cdrom.com` es el *home site* de Slackware.

[4] Quizá cuando lea esto ya contenga una versión mucho más moderna.

2.1.4.2.1 Obtener los ficheros Debería bajarse los siguientes ficheros usando FTP. Asegúrese de poner el modo binario antes de comenzar la transferencia. El Appéndice C contiene un tutorial completo sobre el uso de FTP.

- Los diversos ficheros `README`, y los `SLACKWARE_FAQ`. Asegúrese de leer esos ficheros antes de intentar instalar el software, para ver cualquier actualización o cambio en éstos ficheros.
- Una imagen de bootdisk. Este es un fichero que escribirá en un disquette para crear el bootdisk Slackware. Si tiene una disquetera de arranque de 1.44 megabytes (3.5"), mire en el directorio `bootdsks.144`. Si tiene una disquetera de arranque de 1.2 megabytes (5.25"), mire en el directorio `bootdsks.12`.

  Necesita uno de los siguientes ficheros:

  – `bare.gz`. Este es un bootdisk que sólo tiene drivers para discos duros del tipo IDE. (Sin soporte para SCSI, CD-ROM o de red.) Utilice éste si sólo tiene una controladora IDE y no va a instalar Linux a través de la red o desde CD-ROM.
  – `cdu31a.gz`. Contiene drivers IDE, SCSI y del CD-ROM Sony CDU31A/33A.
  – `mitsumi.gz`. Contiene drivers IDE, SCSI y de los CD-ROM de Mitsumi.
  – `modern.gz`. Un bootdisk experimental con un kernel más moderno y con todos los drivers excepto los de tarjetas de red y el del CD-ROM Sony 535.
  – `net.gz`. Contiene drivers IDE y de red.
  – `sbpcd.gz`. Contiene drivers IDE, SCSI y los del CD-ROM de SoundBlaster Pro/Panasonic.
  – `scsi.gz`. Contiene drivers IDE, SCSI y de CD-ROM SCSI.
  – `scsinet.gz`. Contiene drivers IDE, SCSI, CD-ROM SCSI y de red.
  – `sony535.gz`. Contiene drivers IDE, SCSI y del CD-ROM Sony 535/531.
  – `xt.gz`. Contiene drivers de discos duros IDE y XT.

  Sólo necesita *una* de las anteriores imágenes de bootdisk, dependiendo del hardware que tenga en su sistema.

  La cuestión aquí es que algunos drivers hardware entran en conflicto unos con otros de formas extrañas, de modo que a veces, en vez de intentar arreglar problemas hardware, en su sistema es más fácil usar una imagen de bootdisk con sólo ciertos drivers habilitados. La mayor parte de las ocasiones los usuarios deberían probar con el `scsi.gz` o el `bare.gz`.

- Una imagen de rootdisk. Este es un fichero que escribirá en un disquette para crear el disco de instalación de la Slackware. Como con la imagen de bootdisk, mire en `rootdsks.144` o en `rootdsks.12` dependiendo del tipo de disco de arranque que tenga.

  Necesita uno de los siguientes ficheros:

  – `color144.gz`. El disco de instalación basado en menús en color para disqueteras de 1.44 megabytes. La mayoría de los usuarios deberían usar este rootdisk.

- `umsds144.gz`. Una versión del disco `color144` para instalar el sistema de ficheros UMS-DOS, que permite instalar Linux en un sistema de ficheros MS-DOS. Este método de instalación no se trata aquí en detalle, pero le evitará el tener que reparticionar su disco duro. Se ampliará esto más adelante.
- `tty144.gz`. El disco de instalación para disqueteras de 1.44 megabytes en modo terminal (sin menú en colores). Si el `color144.gz` no funciona en su ordenador pruebe con éste en su lugar.
- `colrlite.gz`. El disco de instalación basado en menús para discos de 1.2 megabytes.
- `umsds12.gz`. Una versión del disco `colrlite` para instalar Linux con el sistema de ficheros UMSDOS. Ver la descripción sobre el `umsds144.gz` un poco más arriba.
- `tty12.gz`. El disco de instalación en modo terminal (sin menú de colores) para discos de 1.2 megabytes. Use este rootdisk si tiene una disquetera de arranque de 1.2 megabytes y si el `colrlite.gz` no funciona en su ordenador.

Al igual que antes, sólo necesita *una* de las imágenes detalladas en el punto anterior, dependiendo del tipo de disquetera de arranque de que disponga.

- `GZIP.EXE`. Este es un ejecutable para MS-DOS del compresor `gzip` usado para comprimir los ficheros de imágenes de bootdisk y de rootdisk (la extensión `.gz` de sus nombres indica eso). Se encuentra en el directorio `install`.

- `RAWRITE.EXE`. Este es un programa MS-DOS que escribirá el contenido de un fichero (como los de las imágenes de boot y de rootdisk) directamente a un disquette sin importar el formato. Lo deberá utilizar para crear los discos de boot y de root. También se encuentra en el directorio `install`.

  Solamente necesitará `RAWRITE.EXE` y `GZIP.EXE` si quiere crear los discos de boot y root desde un sistema MS-DOS. Si tiene acceso a una estación UNIX con disquetera, puede crear los discos desde ahí usando el comando `dd`. Consulte la página man sobre `dd` y pregunte al administrador UNIX de su sistema para que le ayude.

- Los ficheros en los directorios `slakware/a1`, `slakware/a2`, y `slakware/a3`. Estos ficheros conforman el disk set “`A`” de la distribución Slackware. Se requerirán más adelante. Deberá copiar éstos ficheros a discos MS-DOS para instalarlos (o también puede instalarlos desde su disco duro). Por lo tanto, cuando se baje éstos ficheros, mantengalos en directorios separados; no mezcle los ficheros del `a1` con los del `a2`, etc.

  Asegúrese también de que obtiene los ficheros sin los puntos del nombre. Esto es, desde FTP, use el comando “`mget *`” en vez del “`mget *.*`”.

- Los ficheros en los directorios `ap1`, `ap2`, etc., dependiendo de qué sets de discos esté instalando. Por ejemplo, si va a instalar la serie de discos “`X`”, consiga los ficheros en los directorios `x1` a `x5`. Y, al igual que con el set de discos “`A`” más arriba, asegúrese de mantener los ficheros en directorios separados cuando se los baje.

2.1.4.3 Conseguir la Slackware en CD-ROM

Slackware también se puede obtener en CD-ROM. La mayoría de los CD-ROMs Slackware simplemente contienen una copia de los ficheros que aparecen en los archivos de los FTP sites, tal y como se han descrito antes. Por lo tanto, si tiene un CD-ROM Slackware, tiene todos los ficheros que necesita.

Deberá crear un disco boot y otro root de los ficheros que hay en el CD-ROM. Ver la Sección 2.1.4.2.1, más arriba, para la descripción de las imagenes boot y root existentes.

Primero, decida qué imagenes boot y root va a usar. Deberían estar todas en el CD-ROM. Más abajo describiremos cómo crear éstos discos.

2.1.4.4 Métodos de instalación

Slackware proporciona varias maneras distintas para instalar el software. La más popular es instalar desde una partición MS-DOS en su disco duro; otra es instalar desde disquettes MS-DOS creados a partir de los disk set que se haya bajado.

Si tiene la Slackware en un CD-ROM, puede instalar los ficheros directamente desde él. La distribución Slackware Pro, de Morse Telecommunications, le permite instalar la slackware de forma que muchos ficheros se acceden directamente del CD-ROM. Esto puede ahorrar una gran cantidad de espacio de su disco duro, aunque con la desventaja de que la ejecución de ciertas aplicaciones serámás lenta.

2.1.4.4.1 Creando los disquettes de boot y de root Debe crear disquettes a partir de las imágenes del bootdisk y del rootdisk que haya obtenido (o que tenga en CD-ROM), sin importar el tipo de instalación que vaya a hacer.

En un sistema MS-DOS, debe descomprimir las imagenes del bootdisk y del rootdisk usando `GZIP.EXE`. Por ejemplo, si está usando la imagen de bootdisk `bare.gz`, ejecute el comando MS-DOS:

```
C:\> GZIP -D BARE.GZ
```

con lo que se descomprimirá `bare.gz` y tendrá el fichero `bare`. Si está instalando desde CD-ROM, puede copiar la imagen de un bootdisk (como el `bare.gz`) a su disco duro, y ejecutar el `GZIP.EXE` del CD-ROM para descomprimirlo.

De forma similar debe descomprimir la imagen del rootdisk. Por ejemplo, si está usando el rootdisk `color144.gz`, ejecute el comando:

```
C:\> GZIP -D COLOR144.GZ
```

que descomprimirá el fichero y obtendrá el `color144`.

Después, deberá tener dos discos de *alta densidad* formateados en MS-DOS (deben ser del mismo tipo; esto es, si su disco de arranque es de 3.5", ambos disquettes deberán ser de 3.5" y de alta densidad). y deberá usar `RAWRITE.EXE` para escribir las imágenes de boot y de rootdisk a los discos.

Ejecute el comando:

```
C:\> RAWRITE
```

Conteste a las preguntas con el nombre del fichero a escribir (por ejemplo `bare`, o `color144`) y con el disco destino al que escribir (por ejemplo `A:`). `RAWRITE` copiará el fichero, bloque a bloque, directamente al disco. De la misma forma use `RAWRITE` con la imagen del rootdisk. Cuando haya acabado tendrá dos discos: uno será el bootdisk y el otro el rootdisk. Tenga en cuenta que estos dos discos ya no son legibles por MS-DOS (son discos con "formato Linux").

Asegúrese de que esté usando discos nuevos y libres de errores. No deben tener ningún bloque defectuoso.

Tenga en cuenta de que no necesita estar en MS-DOS para instalar la Slackware. Sin embargo, estar en MS-DOS facilita la creación de los discos de boot y de root, y hace más fácil la instalación del software (como el poder instalar directamente desde una partición MS-DOS de su sistema). Si no está utilizando MS-DOS en su sistema, puede usar el sistema MS-DOS de otra persona, sólo para crear los discos, e instalar desde ellos.

No es necesario, tampoco, utilizar `GZIP.EXE` y `RAWRITE.EXE` bajo MS-DOS para crear los discos boot y root. Puede usar los comandos `gzip` y `dd` en un sistema UNIX para llevar a cabo la misma tarea (para esto, necesitará una estación UNIX con disquetera, por supuesto.) Por ejemplo, en una estación SUN con la disquetera en el dispositivo `/dev/rfd0`, puede usar los comandos:

```
$ gunzip bare.gz
$ dd if=bare of=/dev/rfd0 obs=18k
```

Debe proporcionar el argumento con el adecuado tamaño de bloque (el argumento `obs`) en algunas estaciones (por ejemplo, en las Suns) o, si no, no lo hará bien. Si tiene problemas, la página man sobre `dd` puede ser de ayuda.

2.1.4.4.2 Preparándose para la instalación desde disco duro Si lo que quiere es instalar la Slackware directamente desde el disco duro (lo que muchas veces es más rápido y más fiable que una instalación desde disquette), necesitará una partición MS-DOS en el sistema en el que vaya a instalar la Slackware.

Nota: Si va a instalar la Slackware desde una partición MS-DOS, esta partición NO debe estar comprimida con DoubleSpace, Stacker, o cualquier otro tipo de utilidad de compresión de disco. Por el momento Linux no puede leer particiones MS-DOS comprimidas con DoubleSpace o Stacker diréctamente. (Aunque se puede acceder a éstas por medio del Emulador de MS-DOS, ésta no es una opción cuando se instala el software Linux.)

Para preparar la instalación desde disco duro, simplemente cree un directorio en él para almacenar los ficheros de la Slackware. Por ejemplo,

```
C:\> MKDIR SLACKWAR
```

creará el directorio `C:\SLACKWAR` que contendrá los ficheros de la Slackware. Bajo este directorio, deberá crear subdirectorios `A1`, `A2`, etc, para cada set de discos que se haya bajado, usando el comando `MKDIR`. Todos los ficheros del disco `A1` se almacenarán en el directorio `SLACKWAR\A1`, y lo mismo con los demás discos.

2.1.4.4.3 Preparándose para la instalación desde disquette Si desea instalar la Slackware desde disquettes en lugar de desde disco duro, necesitará tener un disquette virgen, formateado en MS-DOS, por cada uno de los discos que se haya bajado. Estos discos deben estar formateados en alta densidad.

El set de discos `A` (discos `A1` a `A3`) pueden ser, discos de 3.5", o bien de 5.25". Sin embargo, el resto de los disk set deben ser discos de 3.5". Por lo tanto, si sólo tiene una disquetera de 5.25", necesitará conseguir una de 3.5" de alguien, de forma que pueda instalar los otros disk set aparte del `A`. (O, si no, puede instalar desde el disco duro, como se ha explicado en la sección previa.)

Para crear los discos, simplemente copie los ficheros de cada directorio de la Slackware en un disquette formateado bajo MS-DOS, usando el comando MS-DOS `COPY`. De este modo,

```
C:\> COPY A1\*.* A:
```

copiará los contenidos del disco `A1` a la disquetera `A:`. Deberá repetir esto con cada disco que se haya bajado.

No necesita modificar o descomprimir los discos de ninguna forma; sólamente necesita copiarlos en disquettes MS-DOS. Será el procedimiento de instalación de la Slackware el que se encargue de ello por Ud.

2.1.4.4.4 Preparándose para la instalación desde CD-ROM Si tiene la Slackware en un CD-ROM, ya está preparado para instalar el software una vez que haya creado los disquettes de boot y de root. El software se instalará directamente desde el CD-ROM.

2.2 Preparación para instalar Linux

Una vez que usted ha obtenido una distribución de Linux, estará preparado para instalar el sistema. Esto supone cierto grado de planificación, sobre todo si en su ordenador se usan actualmente otros sistemas operativos. En las secciones siguientes describiremos cómo preparar la instalación de Linux.

2.2.1 Visión general de la instalación

A pesar de ser diferente cada distribución de Linux, el método utilizado para instalar el software es, en general, como sigue:

1. **Reparticione su(s) disco(s) duro(s).** Si usted tiene instalados otros sistemas operativos, necesitará *reparticionar* los discos con el fin de reservar espacio para Linux. Esto se tratará más adelante en la Sección 2.2.4.

2. **Arranque la instalación de Linux.** Cada distribución de Linux incluye algo para arrancar inicialmente e instalar el software, usualmente un diskette de arranque. Arrancando de esta forma, usted entrará en un programa de instalación para el resto del software, o bien le permitirá seguir instalándolo a mano.

3. **Cree las particiones para Linux.** Después de reparticionar su disco para reservar espacio para Linux, debe crear particiones de Linux en dicho espacio. Esto se realiza con el programa `fdisk`, comentado en la Sección 2.3.3.

4. **Cree los sistemas de ficheros y el espacio de intercambio.** En este momento, debe crear uno o más sistemas de ficheros, utilizados para guardar sus ficheros, en las particiones recién creadas. Además, si piensa usar espacio de intercambio ("swap"), debe crear dicho espacio en una de las particiones para Linux. Esto se trata en las Secciones 2.3.4 y 2.3.5.

5. **Instale los programas en los sistemas de ficheros.** Finalmente, debe instalar el software en sus nuevos sistemas de ficheros. Después de esto, lo que queda es fácil—si todo va bien. Esto se trata en la Sección 2.3.6. Más adelante, en la Sección 2.5, describiremos qué hacer si algo no va bien.

La mayoría de las distribuciones de Linux proporcionan un programa de instalación que le guiará en cada paso de la instalación, y automatiza algunos de esos pasos. Tenga en cuenta que cualquiera de los siguientes pasos pueden estar automatizados o no, dependiendo de la distribución.

La distribución Slackware de Linux, explicada en este libro, solo requiere que reparticione su disco, utilizando `fdisk`, y use el programa `setup` para completar los restantes pasos.

Un consejo importante: Mientras instala Linux, lo mejor que le podemos aconsejar es que *tome notas* durante todo el procedimiento de instalación. Escriba todo lo que haga, todo lo que escriba y todo lo que vea y pueda no parecerle normal. La idea aquí es simple: si usted tiene (¡o cuando tenga!) problemas, usted querrá poder retroceder sus pasos y encontrar qué hizo mal. Instalar Linux no es difícil, pero hay que recordar continuamente muchos detalles. Usted querrá llevar un registro de todos esos detalles para poder probar de otra forma si algo le va mal. Además, guardando sus notas sobre su experiencia instalando Linux le será útil cuando quiera pedir ayuda a otras personas, por ejemplo cuando envíe un mensaje a un grupo de news USENET sobre Linux. Sus notas sobre la instalación son también algo que querrá mostrar a sus nietos algún día.

2.2.2 Conceptos sobre particiones

En general, los discos duros se encuentran divididos en *particiones*, donde cada partición corresponde a un sistema operativo. Por ejemplo, en un disco duro se pueden tener varias particiones—una dedicada a, digamos, MS-DOS, otra a OS/2 y otra a Linux.

Si usted tiene ya otro software instalado en su sistema, puede necesitar cambiar el tamaño de sus particiones con el fin de reservar espacio para Linux. En el espacio reservado se crearán una o más particiones para almacenar el software de Linux y el espacio de intercambio. A este proceso lo llamamos *reparticionar*.

La mayoría de los sistemas MS-DOS utilizan una única partición que ocupa todo el disco. Para MS-DOS, esta partición es accedida como `C:`. Si usted tiene más de una partición, MS-DOS las llamará `D:`, `E:`, y así sucesivamente, de modo que cada partición actúa como si fuera un disco duro independiente.

En el primer sector del disco está el **registro de arranque maestro** junto a la **tabla de particiones**. El registro de arranque (como su nombre indica) se usa para arrancar el sistema. La tabla de particiones contiene información acerca del lugar y el tamaño de cada partición.

Hay tres clases de particiones: **primarias**, **extendidas**, y **lógicas**. De éstas, las más usadas son las primarias. Sin embargo, debido al límite del tamaño de la tabla de particiones, sólo pueden tenerse hasta cuatro particiones primarias en un disco.

La forma de superar este límite de cuatro particiones es usar particiones extendidas. Una partición extendida no tiene datos ella misma; en su lugar, actúa como "soporte" de particiones lógicas. Por lo tanto, se puede crear una partición extendida que ocupe todo el disco, y dentro crear cualquier número de particiones lógicas. Sin embargo, sólo puede tenerse una partición extendida por disco.

2.2.3 Necesidades de reparticionado en Linux

Antes de que expliquemos cómo reparticionar sus discos, necesita tener una idea acerca del espacio que necesita para Linux. Trataremos más adelante el tema de crear las particiones, en la Sección 2.3.3.

En los sistemas UNIX, los ficheros se almacenan en un **sistema de ficheros**, que es esencialmente una zona del disco duro (u otro dispositivo, como un CD-ROM o un diskette) formateado para almacenar ficheros. Cada sistema de ficheros se encuentra asociado con una parte específica del árbol de directorios; por ejemplo, en la mayoría de los sistemas, existe un sistema de ficheros para todos los ficheros del directorio `/usr`, otro para `/tmp`, etc. El **sistema de ficheros raíz** es el principal, que corresponde con el directorio raíz, `/`.

Bajo Linux, cada sistema de ficheros ocupa una partición del disco duro. Por ejemplo, si tenemos un sistema de ficheros para `/` y otro para `/usr`, necesitaremos dos particiones para almacenar ambos sistemas.

Antes de instalar Linux, usted necesitará preparar sistemas de ficheros para almacenar el software de Linux. Por lo menos tiene que tener un sistema de ficheros (el sistema de ficheros raíz), y una partición reservada a Linux. La mayoría de los usuarios de Linux optan por almacenar todos sus ficheros en el sistema de ficheros raíz, pues en la mayor parte de los casos es más fácil de gestionar que tener diferentes sistemas de ficheros y particiones.

Sin embargo, usted puede crear varios sistemas de ficheros para Linux si lo desea—por ejemplo, puede que quiera usar sistemas separados para `/usr` y `/home`. Aquellos lectores con experiencia en administración de sistemas UNIX sabrán como usar diferentes sistemas de ficheros adecuadamente. En el capítulo 4 discutiremos la utilización de varias particiones y sistemas de ficheros.

¿Por qué usar más de un sistema de ficheros? Lo más habitual es por seguridad; si, por alguna razón, uno de sus sistemas de ficheros resulta dañado, los otros normalmente no resultarán afectados. Por otro lado, si usted almacena todos sus ficheros en el sistema de ficheros raíz, y por alguna razón resulta dañado, puede perder todos los ficheros de una vez. Sin embargo, esto no es lo habitual; si usted hace copias de seguridad (*backups*) regularmente, no hay de qué preocuparse.[5]

Otra razón para utilizar varios sistemas de ficheros es repartir el almacenamiento entre varios discos duros. Si usted tiene, digamos, 40 megabytes libres en un disco duro y 50 en otro, usted puede querer crear un sistema de ficheros raíz de 40 megabytes en el primer disco y un sistema `/usr` de 50 megabytes en el otro. Actualmente no es posible que un sistema de ficheros abarque varios discos; si su espacio libre de disco está repartido entre los discos, necesitará utilizar varios sistemas de ficheros para aprovecharlos.

En resumen, Linux requiere por lo menos una partición, para el sistema de ficheros raíz. Si se desea crear varios sistemas de ficheros, se necesitará una partición por cada sistema de ficheros. Algunas distribuciones de Linux crean particiones y sistemas de ficheros de forma automática, de modo que usted no necesita preocuparse demasiado de estos asuntos.

Otra cuestión a considerar cuando se deciden las particiones es el espacio de intercambio (*swap*). Si se desea usar espacio de intercambio en Linux, se tienen dos opciones. La primera es usar un *fichero de intercambio* que existe dentro de uno de los sistemas de ficheros de Linux. Se crea el fichero de intercambio para usarlo como RAM virtual una vez instalado el software. La segunda opción es crear una *partición de intercambio*, una partición reservada exclusivamente como espacio de *swap*. La mayoría de la gente usa una partición para el intercambio en lugar de un fichero.

Cada fichero o partición de intercambio puede ser de hasta 16 megabytes. Si se desea tener más de 16 megabytes de *swap*, se pueden crear varios ficheros o particiones de intercambio—hasta ocho a la vez. Por ejemplo, si se necesitan 32 megabytes de *swap*, se pueden crear dos particiones de 16 megabytes.

La Sección 2.3.4 se ocupa de cómo poner a punto una partición para el intercambio, mientras que lo referente a ficheros de *swap* se trata en el Capítulo 4.

Por lo general, se crearán dos particiones para Linux: una para ser usada como sistema de ficheros raíz, y la otra como espacio de intercambio. Por supuesto, hay otras opciones pero esta es la opción mínima. El espacio de *swap* no es obligatorio en Linux, pero está muy recomendado si posee menos de 16 megabytes de memoria física.

También necesita conocer el espacio requerido para cada partición. El tamaño de los sistemas de ficheros de su sistema Linux depende en gran parte de qué software quiera instalar en él y de la distribución de Linux que esté utilizando. Afortunadamente, la documentación de la distribución le dará una idea del espacio requerido. Un sistema pequeño puede utilizar sólo 20 megabytes o menos; un sistema grande siempre necesitará 80 a 100 megabytes, o más. Tenga en cuenta que hay que añadir a esto el espacio extra para los directorios de usuario, expansiones futuras, etc.

El tamaño de sus particiones de *swap* (debe elegirse una para esto) depende de la RAM virtual que necesite. Lo típico es crear una partición de intercambio del doble de espacio de su RAM física; por ejemplo, si tiene 4 megabytes de RAM, una partición de 8 megabytes suele bastar. Por supuesto, esto es solo una idea—la cantidad de espacio de *swap* que requiere dependerá del software que quiera ejecutar. Si usted tiene una gran cantidad de memoria física (digamos, 16 megabytes o más) puede

[5]El autor utiliza un único sistema de ficheros de 200 megabytes para todos sus ficheros de Linux, y no ha tenido ningún problema (hasta ahora).

que al final no necesite espacio de intercambio.

Nota importante: Debido a las limitaciones de la BIOS, habitualmente no es posible arrancar desde particiones que empiecen más allá del cilindro 1023. Así pues, cuando se decida el espacio para Linux, recuerde que no debe usar una partición que sobrepase ese cilindro para el sistema de ficheros raíz. Linux puede *usar perfectamente* particiones con cilindros más allá del 1023, pero dichas particiones no servirán para *arrancar*. Decir esto aquí puede parecer prematuro, pero es importante a la hora de planificar el uso del disco en Linux.

Si se usara una partición para el sistema de ficheros raíz más allá del cilindro 1023, siempre podrá arrancar Linux con un disco *boot*. Esto no es un inconveniente—solo tarda unos segundos más que si se arrancara desde el disco duro.

2.2.4 Reparticionado de los discos

En esta sección vamos a describir cómo cambiar el tamaño de sus particiones actuales (si las hay) para reservar espacio para Linux. Si usted está instalando Linux en un disco duro "limpio", puede pasar directamente a la Sección 2.3 de más adelante.

La manera habitual de cambiar el tamaño de una partición es borrarla (lo que implica borrar toda la información que contenga) y rehacerla. Antes de reparticionar sus discos, *haga un backup*. Después de cambiar las particiones, puede proceder a reinstalar el software desde el *backup*. Sin embargo, podemos encontrar programas para MS-DOS que consiguen cambiar el tamaño de las particiones de forma no destructiva. Uno de éstos se conoce como "`FIPS`", y puede encontrarse en muchos servidores de FTP de Linux.

Recuerde también que debido a que se empequeñecen las particiones originales, no se va a poder reinstalar todo el software que se tenía antes. En este caso, hay que borrar el software innecesario para permitir que el resto quepa en las particiones más pequeñas.

El programa utilizado para hacer particiones es `fdisk`. Cada sistema operativo tiene su propia versión de este programa; por ejemplo, bajo MS-DOS, se activa con el comando `FDISK`. Debe consultarse la documentación de los sistemas operativos en uso para obtener información sobre este asunto. Aquí nos referiremos unicamente a MS-DOS con `FDISK`, pero lo que digamos puede ser fácilmente extrapolado a otros sistemas operativos.

Consulte, *por favor*, la documentación de sus sistemas operativos actuales antes de proceder a reparticionar sus discos. Esta sección pretende ser sólo una visión general del proceso; y hay muchos detalles que no vamos a mencionar. Tenga en cuenta que se puede perder todo el software de su sistema si no se hace correctamente el reparticionado.

Un aviso: No modifique o cree particiones para otros sistemas operativos (incluyendo Linux) utilizando `FDISK` bajo MS-DOS. Solo pueden modificarse particiones de cada sistema operativo con la versión de `fdisk` correspondiente a ese sistema; por ejemplo, se crearán las particiones para Linux utilizando el programa `fdisk` que viene con Linux. Más adelante, en la Sección 2.3.3, describiremos cómo crear particiones de Linux, pero por ahora nos ocuparemos solo de cambiar el tamaño de las actuales.

Supongamos que usted tiene un solo disco duro en su sistema, dedicado, por ahora, enteramente a MS-DOS. Esto es, su disco duro contiene una partición MS-DOS, conocida habitualmente como "`C:`". Puesto que este método de reparticionado destruirá todos los datos de la partición, necesita crear un disco de sistema MS-DOS "arrancable" que contenga lo necesario para ejecutar `FDISK` y restaurar el software desde el backup cuando se complete el proceso de reparticionado.

En muchos casos, se pueden usar para esto los discos de instalación de MS-DOS. Sin embargo, si necesita el disco de sistema, formatéelo mediante el comando

```
FORMAT /s A:
```

Copie en ese disco todas las utilidades de MS-DOS necesarias (normalmente, casi todo lo que hay en el directorio `\DOS` de su disco), así como los programas `FORMAT.COM` y `FDISK.EXE`. Ahora se debe poder arrancar desde este disquete, y ejecutar el comando

```
FDISK C:
```

para arrancar `FDISK`.

La utilización de `FDISK` debería ser autoexplicativa, pero puede consultarse la documentación de MS-DOS para obtener detalles. Cuando comience el programa `FDISK`, utilice el menú de opciones para mostrar la tabla de particiones, y *anote* la información que se le muestre. Es importante guardar copia de su configuración original en caso de que quiera detener la instalación de Linux.

Para borrar una partición, seleccione la opción del menú "`Delete an MS-DOS Partition or Logical DOS Drive`" (*Eliminar partición o unidad lógica DOS*). Especifique el tipo de partición que desea borrar (primaria, extendida o lógica) y el número de la partición. Verifique todos los avisos que se le presenten.

Para crear una nueva partición para MS-DOS (más pequeña), seleccione la opción de `FDISK` "`1. Crear partición DOS o unidad lógica DOS`". Especifique el tipo de partición (primaria, extendida o lógica) y el tamaño (en megabytes). `FDISK` deberá crear la partición.

Después de hacer esto mediante `FDISK`, debe abandonar el programa y reformatear las nuevas particiones. Por ejemplo, si se cambió el tamaño de la partición `C:` se tecleará el comando

```
FORMAT /s C:
```

Ahora puede reinstalar el software desde el backup.

2.3 Instalación del software de Linux

Después de modificar sus particiones para reservar espacio a Linux, usted está preparado para instalar el software. A continuación se muestra un resumen del procedimento a seguir:

- Arrancar con el dispositivo de arranque de Linux (disquete);
- Ejecutar `fdisk` bajo Linux para crear las particiones de Linux;

- Ejecutar `mke2fs` y `mkswap` para crear los sistemas de ficheros y el espacio de intercambio;
- Instalar el software de Linux;
- Finalmente, instalar el cargador `LILO` en el disco duro, o crear un disco de arranque con el fin de arrancar su nuevo sistema Linux.

Como hemos dicho, uno (o más) pasos de los anteriores pueden estar automatizados por los programas de instalación, según la distribución que esté utilizando. Por favor, consulte la documentación de su distribución para ver las instrucciones específicas.

2.3.1 Arranque de Linux

El primer paso es iniciar el computador con el dispositivo de arranque de Linux, que suele ser un disco "boot" que contiene un pequeño sistema Linux. Tras arrancar con el floppy, se le presentará un menú de instalación de algún tipo que le guiará en el proceso de instalación. En otras distribuciones, se le mostrara un *prompt* de *login* cuando arranque. Aquí se suele entrar como `root` o `install` para comenzar el proceso de instalación.

La documentación que viene con su particular distribución le explicará qué necesita para arrancar Linux.

Si usted está instalando la distribución de Linux *Slackware*, lo único requerido es arrancar con el disco "boot" que creó en la sección anterior.

La mayoría de las distribuciones de Linux utilizan un disquete de arranque que le permite introducir parámetros de su hardware en tiempo de arranque, para forzar la detección de sus dispositivos. Por ejemplo, si su controladora SCSI no se detecta durante el arranque, necesitará rearrancar y especificar los parámetros del hardware (como direcciones E/S e IRQ) en el prompt de arranque.

Asímismo, las máquinas PS/1, ThinkPad y ValuePoint de IBM no almacenan la información de geometría de los discos en la CMOS, con lo que debe especificarla durante el arranque.

El prompt de arranque se muestra siempre que se arranca con el disquete. Este es el caso de la distribución Slackware. En otras es necesario mantener pulsadas las teclas `shift` o `ctrl` mientras se arranca. Si va bien, debería verse el prompt

```
boot:
```

y tal vez otros mensajes.

Para arrancar sin más parámetros especiales, pulse `enter` en el prompt del arranque.

Observe los mensajes del arranque. Si tiene una controladora SCSI, debería ver una lista de hosts SCSI detectados. Si observa el mensaje

```
SCSI: 0 hosts
```

es porque no se detectó su controladora SCSI, y tendrá que seguir el siguiente procedimiento.

Además, el sistema le informará de las particiones y dispositivos detectados. Si cualquier parte de esta información es incorrecta (o falta alguno), tendrá que forzar la detección del hardware.

Por otro lado, si todo va bien y todo su hardware es correctamente detectado, puede pasar a la siguiente sección, la Sección 2.3.2.

Para forzar la detección del hardware, debe entrar los parámetros adecuados en el prompt de arranque, utilizando la siguiente sintaxis:

`ramdisk` ⟨*parameters...*⟩

Hay cierto número de parámetros disponibles; aquí mostraremos los más comunes.

`hd=`⟨*cylinders*⟩`,`⟨*heads*⟩`,`⟨*sectors*⟩
: Especifica la geometría del disco. Requerido para sistemas como el IBM PS/1, ValuePoint y ThinkPad. Por ejemplo, si su disco tiene 683 cilindros, 16 cabezas y 32 sectores por pista, introduzca

```
ramdisk hd=683,16,32
```

`tmc8xx=`⟨*memaddr*⟩`,`⟨*irq*⟩
: Especifica las direcciones e IRQ para el controlador SCSI Future Domain TMC-8xx. Por ejemplo,

```
ramdisk tmc8xx=0xca000,5
```

Observe que el prefijo `0x` debe utilizarse para todos los valores que se dan en hexadecimal. Esto se cumple con todas las opciones siguientes.

`st0x=`⟨*memaddr*⟩`,`⟨*irq*⟩
: Especifica las direcciones e IRQ para el controlador Seagate ST02.

`t128=`⟨*memaddr*⟩`,`⟨*irq*⟩
: Especifica las direcciones e IRQ para el controlador Trantor T128B.

`ncr5380=`⟨*port*⟩`,`⟨*irq*⟩`,`⟨*dma*⟩
: Especifica el puerto, IRQ y canal DMA para el controlador genérico NCR5380.

`aha152x=`⟨*port*⟩`,`⟨*irq*⟩`,`⟨*scsi_id*⟩`,1`
: Especifica puerto, IRQ e identificador SCSI para controladores AIC-6260. Esto incluye a los controladores Adaptec 1510, 152x y Soundblaster-SCSI.

Para cada uno de éstos, debe entrar `ramdisk` seguido del parámetro que desee utilizar.

Si tiene dudas acerca de estas opciones de arranque, por favor lea el documento Linux *SCSI HOWTO*, que debe estar disponible en cualquier *FTP-site* de Linux (o donde haya obtenido este libro), así como el documento Linux *CD-ROM HOWTO*. Estos documentos describen cuestiones de compatibilidad de hardware con mucho más detalle.

2.3.2 Dispositivos y particiones en Linux

Muchas distribuciones necesitan que se creen a mano las particiones de Linux utilizando el programa `fdisk`. Otras pueden crearlas automáticamente. En cualquier caso, usted debe conocer lo siguiente acerca de los nombres para los dispositivos y las particiones en Linux.

Bajo Linux, los dispositivos y las particiones tienen nombres muy distintos a los utilizados en otros sistemas operativos. Bajo MS-DOS, las disqueteras se identifican como `A:` y `B:`, mientras que las particiones del disco duro se identifican como `C:`, `D`, etc. Bajo Linux, la denominación es algo diferente.

Los **manejadores de dispositivos**, que se encuentran en el directorio `/dev`, se usan para comunicar con los dispositivos de su sistema (como discos duros o ratones). Por ejemplo, si usted tiene un ratón en su sistema, lo puede acceder a través del manejador `/dev/mouse`. Las disqueteras, discos duros y particiones tienen cada uno un manejador propio. No se preocupe acerca de la interfaz del manejador por ahora; solo es importante entender cómo son nombrados los dispositivos con el fin de poderlos usar.

La tabla 2.1 lista los nombres de diversos manejadores.

Dispositivo	Nombre
Primera disquetera (`A:`)	`/dev/fd0`
Segunda disquetera (`B:`)	`/dev/fd1`
Primer disco duro (*todo* el disco)	`/dev/hda`
Primer disco duro, partición primaria 1	`/dev/hda1`
Primer disco duro, partición primaria 2	`/dev/hda2`
Primer disco duro, partición primaria 3	`/dev/hda3`
Primer disco duro, partición primaria 4	`/dev/hda4`
Primer disco duro, partición lógica 1	`/dev/hda5`
Primer disco duro, partición lógica 2	`/dev/hda6`
⋮	
Segundo disco duro (*todo* el disco)	`/dev/hdb`
Segundo disco duro, partición primaria 1	`/dev/hdb1`
⋮	
Primer disco duro SCSI (*todo* el disco)	`/dev/sda`
Primer disco duro SCSI, partición primaria 1	`/dev/sda1`
⋮	
Segundo disco duro SCSI (*todo* el disco)	`/dev/sdb`
Segundo disco duro SCSI, partición primaria 1	`/dev/sdb1`
⋮	

Tabla 2.1: Nombres de discos y particiones en Linux

Algunas notas acerca de esta tabla. Observe que `/dev/fd0` corresponde a la primera disquetera (`A:` bajo MS-DOS) y que `/dev/fd1` corresponde a la segunda (`B:`).

Además, los discos duros SCSI se nombran de manera diferente a otros discos. Los IDE, MFM y RLL se acceden a través de los dispositivos `/dev/hda`, `/dev/hdb`, etc. Las particiones de `/dev/hda` son `/dev/hda1`, `/dev/hda2`, etc. Sin embargo, los dispositivos SCSI se nombran con `/dev/sda`, `/dev/sdb`, etc., y las particiones con `/dev/sda1`, `/dev/sda2`, etc.

Aquí tenemos un ejemplo. Supongamos que usted tiene un disco duro IDE con 3 particiones primarias. Las dos primeras son para MS-DOS, y la tercera es extendida y contiene dos particiones lógicas, ambas para ser usadas con Linux. Los dispositivos quedarían representados con:

Primera partición MS-DOS (`C:`)	`/dev/hda1`
Segunda partición MS-DOS (`D:`)	`/dev/hda2`
Partición extendida	`/dev/hda3`
Primera partición lógica de Linux	`/dev/hda5`
Segunda partición lógica de Linux	`/dev/hda6`

Observe que nos hemos saltado `/dev/hda4`, ya que corresponde a la cuarta partición primaria, que no existe en el ejemplo. Las particiones lógicas se nombran de forma consecutiva partiendo de `/dev/hda5`.

2.3.3 Creación de las particiones en Linux

Ahora ya está preparado para crear las particiones de Linux con el comando `fdisk`. Como se explicó en la Sección 2.2.3, va a tener que crear, en general, una partición para el software de Linux propiamente dicho y otra para el área de intercambio.

Después de arrancar el disquete, ejecute el comando `fdisk` tecleando

`fdisk` ⟨*drive*⟩

donde ⟨*drive*⟩ es el nombre de dispositivo con el que Linux identifica el disco duro donde quiere realizar las particiones (véase la Tabla 2.1). Por ejemplo, si desea ejecutar `fdisk` sobre el primer disco SCSI de su sistema, utilice el comando `fdisk /dev/sda`. Por defecto, `fdisk` actúa sobre `/dev/hda` (el primer disco IDE).

Para crear particiones de Linux en más de un disco, ejecute `fdisk` una vez por disco.

```
# fdisk /dev/hda
Command (m for help):
```

En este punto, `fdisk` está esperando un comando; puede teclear `m` para obtener una lista de opciones.

```
Command (m for help): m
Command action
a toggle a bootable flag
d delete a partition
l list known partition types
m print this menu
n add a new partition
```

```
p print the partition table
q quit without saving changes
t change a partition's system id
u change display/entry units
v verify the partition table
w write table to disk and exit
x extra functionality (experts only)

Command (m for help):
```

El comando `n` se usa para crear una nueva partición. Casi todas las demás opciones no nos van a preocupar ahora mismo. Para salir de `fdisk` sin salvar cambios, utilice el comando `q`. Para salir escribiendo los cambios en la tabla de particiones, utilice el comando `w`.

Lo primero que debe hacer es mostrar su tabla de particiones actual y anotar sus datos, para referencias posteriores. Use el comando `p` para esto.

```
Command (m for help):  p
Disk /dev/hda:  16 heads, 38 sectors, 683 cylinders
Units = cylinders of 608 * 512 bytes

   Device Boot  Begin   Start     End  Blocks   Id  System
/dev/hda1   *       1       1     203   61693    6  DOS 16-bit >=32M


Command (m for help):
```

En este ejemplo, tenemos una partición única en `/dev/hda1`, con 61693 bloques (unos 60 megabytes).[6] Esta partición comienza en el cilindro 1 y finaliza en el 203. En total el disco tiene 683 cilindros de los cuales 480 están libres para crear particiones de Linux.

Para crear una nueva partición, utilice el comando `n`. En este ejemplo crearemos dos particiones primarias (`/dev/hda2` y `/dev/hda3`) para Linux.

```
Command (m for help):  n
Command action
e extended
p primary partition (1-4)
p
```

Aquí, `fdisk` pide el tipo de partición a crear: extendida o primaria. En nuestro ejemplo elegimos `p` pues solo vamos a crear particiones primarias.

```
Partition number (1-4):
```

`fdisk` preguntará entonces por el número de la partición a crear; puesto que la 1 está en uso, nuestra primera partición para Linux debe ser la 2.

[6]En Linux, un bloque son 1024 bytes.

```
Partition number (1-4):  2
First cylinder (204-683):
```

Ahora debe entrar el cilindro de comienzo de la partición. Dado que actualmente no están en uso los cilindros 204 a 683, escogeremos el primero disponible (204), ya que no hay razón para dejar huecos entre particiones.

```
First cylinder (204-683):  204
Last cylinder or +size or +sizeM or +sizeK (204-683):
```

Ahora `fdisk` nos está preguntando acerca del tamaño de la partición a crear. Podemos hacerlo especificando el cilindro de terminación de la partición o metiendo directamente el tamaño requerido, en bytes, kilobytes, o megabytes. Como queremos que la partición ocupe 80 megabytes, especificaremos `+80M`. Cuando se indica el tamaño de esta forma, `fdisk` lo redondea a un número de cilindros.

```
Last cylinder or +size or +sizeM or +sizeK (204-683):  +80M
Warning:  Linux cannot currently use 33090 sectors of this partition
```

Si usted ve un mensaje como el anterior, puede ignorarlo. `fdisk` imprime este aviso debido a que es un programa antiguo que data de cuando las particiones de Linux no podían superar los 64 megabytes.

Ahora podemos pasar a crear la segunda partición. Como ejemplo, vamos a crearla de 10 megabytes.

```
Command (m for help):  n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4):  3
First cylinder (474-683):  474
Last cylinder or +size or +sizeM or +sizeK (474-683):  +10M
```

Finalmente, vamos a ver la tabla de particiones. Una vez más, anote la información que se le presente—sobre todo los tamaños en bloques de las nuevas particiones. Necesitará conocerlos cuando tenga que crear, más tarde, los sistemas de ficheros. Además, debe verificar que las particiones no se solapen.

```
Command (m for help):  p
Disk /dev/hda:  16 heads, 38 sectors, 683 cylinders
Units = cylinders of 608 * 512 bytes

   Device Boot  Begin   Start     End  Blocks   Id  System
/dev/hda1   *       1       1     203   61693    6  DOS 16-bit >=32M
```

```
/dev/hda2          204      204      473    82080    81  Linux/MINIX
/dev/hda3          474      474      507    10336    81  Linux/MINIX
```

Como puede ver, ahora en `/dev/hda2` tenemos una partición de 82080 bloques (aproximadamente 80 megabytes) y en `/dev/hda3` tenemos 10336 bloques (unos 10 megabytes).

Observe que muchas distribuciones (como la Slackware) requieren que se utilice el comando `t` en `fdisk` para cambiar el tipo de la partición elegida para el intercambio a "Linux swap", normalmente con el número 82. Se puede usar el comando `L` para ver una lista de tipos de particiones conocidas, y luego `t` para establecer el tipo de la partición de intercambio a "Linux swap".

De esta forma, el software de instalación podrá encontrar automáticamente sus particiones de *swap* en función del tipo. Si el software de instalación no pudiera reconocer su partición de intercambio, deberá repetir la ejecución de `fdisk` y utilizar el comando `t` sobre la partición en cuestión.

En el ejemplo anterior, los cilindros sobrantes (508 a 683) se quedan sin usar. Puede que se desee hacerlo así, para más adelante crear más particiones.

Finalmente, utilizaremos el comando `w` para escribir los cambios en el disco y salir.

```
Command (m for help):  w
#
```

Recuerde que ningún cambio hecho durante la ejecución de `fdisk` tendrá efecto hasta que se teclee el comando `w`, por lo que se puede jugar con diferentes configuraciones y salvarla solo cuando se esté decidido. Además, se puede usar el comando `q` para abandonar `fdisk` sin hacer ningún cambio. Recuerde también que las particiones de otros sistemas operativos no deben tocarse desde el programa `fdisk` de Linux.

Asímismo, recuerde que no se puede arrancar Linux desde una partición que comience más allá del cilindro 1023. Por lo tanto, se puede crear la partición de raíz en el rango inferior a este cilindro o, si esto es imposible, arrancar siempre desde un disquete.

Algunas distribuciones de Linux necesitan rearrancar el sistema tras ejecutar `fdisk`. Esto permite que los cambios en la tabla de particiones tengan efecto. Las nuevas versiones de `fdisk` cambian de forma automática esta información en el núcleo, con lo que no es necesario rearrancar. Lo más seguro es volver a arrancar tras crear las particiones.

2.3.4 Creación del espacio de intercambio (*swap*)

Si usted está pensando en usar una partición de intercambio para obtener RAM virtual, es ahora el momento de hacerlo.[7] En el Capítulo 4 discutiremos la preparación de un fichero de *swap* para el caso de que no desee usar una partición para esto.

Muchas distribuciones necesitan que se cree y active la partición de intercambio antes de instalar el software. Si usted tiene poca RAM física, la instalación puede no ir bien, a menos que se active una cierta cantidad de *swap*.

[7] De nuevo, algunas distribuciones de Linux preparan el área de intercambio automáticamente, o bien mediante un menú de instalación.

La distribución Slackware necesita que se cree el área de *swap* antes de la instalación, si se tienen 4 megabytes o menos. Si este no es el caso, el procedimiento de instalación de Slackware puede usarse para preparar la partición de intercambio automáticamente. Si no está seguro, siga con el procedimiento descrito aquí.

El comando utilizado para preparar una partición de intercambio es `mkswap`, tecleándose

`mkswap -c` ⟨*partition*⟩ ⟨*size*⟩

donde ⟨*partition*⟩ es el nombre de la partición de *swap* y ⟨*size*⟩ es el tamaño de la partición, en bloques.[8] Por ejemplo, si su partición de intercambio es la `/dev/hda3` y tiene 10336 bloques, teclee el comando

`#` *mkswap –c /dev/hda3 10336*

La opción `-c` indica a `mkswap` que compruebe si hay bloques erróneos en la partición mientras la crea.

Si se usan varias particiones de intercambio, se necesitará ejecutar el comando `mkswap` apropiado para cada partición.

Después de preparar el área de *swap*, hay que decirle al sistema que la use. Normalmente, el sistema comienza a usarla automáticamente durante el arranque. Sin embargo, como aun no tiene instalado el software de Linux, tiene que activarla a mano.

El comando para hacerlo es `swapon`, y tiene el formato

`swapon` ⟨*partition*⟩

En el ejemplo anterior, para activar el espacio de intercambio en `/dev/hda3`, usaremos el comando

`#` *swapon /dev/hda3*

2.3.5 Creación de los sistemas de ficheros

Antes de que se puedan usar las particiones de Linux para almacenar ficheros, hay que crear los **sistemas de ficheros** en ellas. La creación de un sistema de ficheros es análoga a formatear una partición de MS-DOS u otros sistemas operativos. Ya discutimos los sistemas de ficheros en la Sección 2.2.3.

Hay varios tipos de sistemas de ficheros disponibles en Linux. Cada tipo de sistema de ficheros tiene su propio formato y características (como longitud del nombre de los ficheros, tamaño máximo, etc). Además, Linux soporta sistemas de ficheros "de terceros" como el de MS-DOS.

El tipo de sistema de ficheros más usado es el **Sistema de Ficheros Extendido 2**, o ext2fs. El *ext2fs* es uno de los más eficientes y flexibles sistemas; permite hasta 256 caracteres en los nombres

[8] Este tamaño es dado por `fdisk`, utilizando la opción `p`. Un bloque en Linux son 1024 bytes.

de los ficheros y tamaños de éstos de hasta 4 Terabytes. En el Capítulo 4 veremos los diferentes tipos disponibles en Linux. De momento nos conformaremos con sugerirle el sistema *ext2fs*.

Si usted está instalando la distribución Slackware, el propio procedimiento de instalación descrito en la próxima sección crea los sistemas de ficheros de forma automática. Si desea crear sus propios sistemas a mano, siga el método que a continuación describimos.

Para crear un sistema tipo *ext2fs* utilice el comando

`mke2fs -c` ⟨*partition*⟩ ⟨*size*⟩

donde ⟨*partition*⟩ es el nombre de la partición, y ⟨*size*⟩ es el tamaño de la partición en bloques. Por ejemplo, para crear un sistema de 82080 bloques en `/dev/hda2`, use el comando

`#` *mke2fs -c /dev/hda2 82080*

Si quiere usar varios sistemas de ficheros en Linux, necesitará repetir el comando `mke2fs` por cada sistema de ficheros.

Si se ha encontrado con algún problema en este punto, lea la Sección 2.5 al final de este capítulo.

2.3.6 Instalación del software

Finalmente, ya está preparado para instalar el software en su sistema. Cada distribución tiene una forma distinta de hacerlo. Muchas tienen un programa que le guía paso a paso en este proceso. En otras, usted tendrá que **montar** sus sistemas de ficheros en un directorio (como `/tmp`) y copiar el software a éste a mano. En las distribuciones en CD-ROM puede seguir la opción de instalar una parte de lo que contiene en su disco duro y dejar el resto (la mayor parte) en el CD-ROM.

Algunas distribuciones ofrecen diversos mecanismos para instalar el software. Por ejemplo, puede instalarlo directamente desde una partición MS-DOS de su disco duro, en lugar de hacerlo desde los disquetes. O incluso puede hacerlo a través de una red TCP/IP mediante FTP o NFS. Consulte la documentación de la distribución para ver detalles.

Por ejemplo, la distribución Slackware sólo necesita de usted que cree las particiones con `fdisk`, y el espacio de intercambio con `mkswap` y `swapon` (si tiene 4 megabytes o menos de RAM), y a continuación ejecute el programa `setup`, que le guía mediante un menú bastante autoexplicativo en la instalación del software. La utilización de `setup` se describe en detalle luego.

El método exacto para instalar el software de Linux difiere en gran parte según la distribución. Confiamos en que instalar el software de Linux es autoexplicativo, como sucede con la mayoría de las distribuciones.

2.3.6.1 Instalando la Slackware con `setup`

Si está instalando la Slackware, después de crear las particiones (y probablemente de crear espacio para swap), utilice el comando

```
# setup
```

Esto le llevará a un menú que le dirigirá a través del resto de los pasos de la instalación.

El procedimiento descrito aquí corresponde con el que se encuentra en los rootdisks `color144` y `colrlite`; los otros rootdisks pueden seguir procesos ligeramente diferentes.

El menú del `setup` consta de los siguientes elementos. Utilice los cursores para moverse por las diferentes opciones, y presione `enter` ó `spacebar` para seleccionar uno.

`Help` Para ver el fichero de ayuda del `setup`.

`Keymap` Esta opción le permite especificar el mapa de teclado para su sistema si no tiene un teclado USA. Se le presentará una lista de mapas de teclado; seleccione el elemento más apropiado de la lista.

`Quick` Esta le permite seleccionar entre los modos “quick” y “verbose”. El modo “Verbose” es el modo por defecto, y está recomendado para la mayoría de las instalaciones (salvo que haya instalado la Slackware una docena de veces, en cuyo caso ya se lo sabrá de memoria).

`Make tags` Esto permite a los expertos en instalaciones de la Slackware crear “tag files” (“ficheros etiqueta”) personalizados, para preseleccionar paquetes. Esto sólo es necesario para personalizar el procedimiento de instalación; y no debería preocuparse de ello.

`Addswap` Este será el primer punto que la mayor parte de los usuarios seleccionará para instalar la Slackware. Se presentará una lista con las particiones de swap disponibles (las particiones con tipo “Linux swap” fijadas desde el `fdisk`), y podrá especificar qué particiones desea utilizar para espacio de swap. Después se le preguntará si desea ejecutar `mkswap` en esas particiones.

Si ya ha ejecutado `mkswap` y `swapon` (como se describió en la Sección 2.3.4) en sus particiones swap, entonces *no* debería permitir al `setup` ejecutar `mkswap` en esas particiones.

De todos modos, incluso aunque ya haya ejecutado `mkswap` y `swapon`, es necesario el uso del elemento `Addswap` del menú: esto asegura que las particiones swap estarán disponibles una vez que tenga el sistema instalado.

◇ ¡Cuidado! Crear espacio de swap en una partición destruirá todos los datos de esa partición. Asegúrese de que no está borrando información que quiera conservar.

Si selecciona éste elemento del menú, se le preguntará automáticamente si desea seguir con los siguientes pasos. En general, siempre deberá contestar afirmativamente.

`Target` Este elemento permite especificar las particiones en las que se va a instalar Linux. Se presentará una lista con las particiones disponibles (aquellas que se especificaron como de tipo “Linux native” en el `fdisk`), y se pedirá que introduzca el nombre de la partición raiz de Linux, como por ejemplo `/dev/hda2`. Después se preguntará por el tipo de sistema de ficheros que desea crear; nosotros sugerimos utilizar el tipo de sistema de ficheros `ext2fs` como se describió en la Sección 2.3.5. Esto creará un sistema de ficheros en la partición nombrada—algo parecido a formatear una partición bajo MS-DOS.

También se preguntará por cualquier otra partición que quiera utilizar con Linux. Por ejemplo, si crea una partición separada para `/usr` (mirar Sección 2.2.3), podrá especificar el nombre de la partición y la localización donde se vaya a montar (como por ejemplo, en `/usr` o `/usr/bin`).

◇ ¡Precaución! Crear un sistema de ficheros en una partición destruirá todos los datos de esa partición. Asegurese de que no borra datos que quiera conservar.

Incluso aunque ya haya creado sistemas de ficheros usando `mke2fs` (ver Sección 2.3.5), deberá usar el elemento del menú `Target` para especificar las particiones donde será instalado el Linux.

`Source` Con ésto se podrá especificar el lugar desde donde se instalará la Slackware, sea disquette, disco duro, o CD-ROM.

Si está instalando desde disco duro, se le preguntará en qué partición se encuentran los archivos de la Slackware, y el tipo de partición de que se trata. Por ejemplo, si tiene los ficheros de la Slackware en una partición MS-DOS, introduzca el nombre de la partición (como por ejemplo `/dev/hda1`) y seleccione `MS-DOS FAT` como el tipo. Después se le preguntará en qué directorio dentro de esa partición se encuentran los ficheros. Por ejemplo, si tiene los ficheros de la Slackware almacenados en el directorio `C:\SLACK` en su partición MS-DOS, introduzca

```
/slack
```

como el lugar de almacenamiento. Dese cuenta de que debe usar barras '/' y no '\', en el nombre del directorio.

Si está instalando desde CD-ROM, se le preguntará por el tipo de CD-ROM que esté usando y por el directorio dentro del CD-ROM en el que se encuentren los ficheros. Muchos CD-ROMs tienen los ficheros en el directorio `/slakware`, pero ésto ya depende de la edición que tenga.

Si está instalando Slackware Professional,[9] se utilizan dos directorios del CD-ROM. `slakware` se usa para el sistema estándar, que instalará los ficheros diréctamente en el disco duro. Y `slackpro` se usa para el sistema basado en CD-ROM, en el que muchos de los ficheros se acceden diréctamente desde el CD-ROM. Esto puede ahorrar espacio de disco, pero el acceso a muchos ficheros es notablemente más lento. Otros distribuidores de la Slackware también permiten la posibilidad de ejecutar el software desde el CD-ROM. De todos modos, si tiene espacio en disco disponible, es recomendable no ejecutar la Slackware desde el propio CD-ROM. Generalmente el rendimiento es más bajo.

[9] Slackware Professional es una versión de la Slackware distribuida por Morse Telecommunications.

Si intenta una instalación desde disco duro o desde CD-ROM, la Slackware puede advertirle de que hay un error de `mount` en éste punto. Esto generalmente es una indicación de que ha habido un problema accediendo al disco duro o al CD-ROM. Consulte la Sección 2.5.3 para más información si se encuentra con un mensaje de este tipo.

`Disk sets` Esta opción del menú permite seleccionar los sets de discos que desee instalar. Como mínimo debe instalar el disk set `A`. Simplemente debe usar los cursores y la barra espaciadora para seleccionar los sets de discos que quiera instalar.

Observe que seleccionar un set de discos no significa que vayan a ser instalados todos los paquetes de ese set de discos; se le preguntará antes de instalar los paquetes marcados como "optional" o "recommended" en el set de discos.

`Install` Por fin, esta opción del menú instalará el software en su sistema. Se le preguntará por el modo de preguntas que se le harán. La mayor parte de los usuarios deberán seleccionar "normal". Para cada set de discos que haya seleccionado, los paquetes marcados como "required" serán instalados, y se le preguntará cuando se vayan a instalar los paquetes marcados "optional" y "recommended". Si está instalando desde disquette se le irá preguntando por los disquettes en sucesión para que los vaya insertando.

Mientras se esté instalando cada paquete aparecerá en pantalla una breve descripción de su contenido. Salvo que tenga algún conocimiento de UNIX o de Linux, muchas de estas descripciones no tendrán mucho significado para Ud. Tome nota de que paquetes están siendo instalados, de forma que sepa lo que contienen, pero no se preocupe de intentar entender todo lo que aparezca en la pantalla.

El error más común con el que se puede encontrar es que no se localice un fichero en un disquette, o un error de E/S al intentar leer del disquette. Esto será una indicación de que los ficheros del disquette pueden estar corruptos o incompletos; o que el disquette en sí está mal. Cualquier disquette que dé ese error deberá ser reemplazado, y deberá reinstalar el set de discos que contenga esos ficheros. Consulte la Sección 2.5.3 para más ayuda.

También es posible que obtenga errores al intentar acceder al CD-ROM; asegúrese de que el CD-ROM está limpio, sin marcas de dedos, etc.

`Configure` Esta opción realiza algunas configuraciones post-instalación en el sistema. Se tratarán en la siguiente sección.

2.3.7 Creación del disco de arranque o instalación del LILO

Cada distribución proporciona mecanismos para arrancar Linux cuando ya esté instalado en su sistema. En la mayoría de los casos se creará un disquete "boot" que contiene el núcleo de Linux configurado para usar con su recién creado sistema de ficheros raíz. Para arrancar Linux, deberá hacerlo desde ese disquete y tras el arranque se pasará el control a su disco duro. En otras distribuciones, el disco de arranque es el propio disquete de instalación.

La mayoría de las distribuciones le van a dar la opción de instalar **LILO** en su disco duro. LILO es un programa que se instala en el registro maestro de arranque del disco, y está preparado para arrancar varios sistemas operativos, entre los que se incluyen MS-DOS y Linux, permitiéndole elegir qué sistema quiere arrancar en cada momento.

En el caso de la distribución Slackware, la opción `Configure` del menú `setup` le permitirá tanto crear un disquete de arranque como instalar LILO. Esas opciones deberían ser suficientemente auto-explicativas. El menú `Configure` también sirve para especificar el módem y el ratón de que dispone, así como su zona horaria.

Con el fin de instalar LILO correctamente, necesita conocer bastante información acerca de la configuración de su disco—por ejemplo, qué particiones contiene cierto sistema operativo, cómo arrancar cada sistema operativo, etc. La mayoría de las distribuciones, cuando se instala LILO, tratan de elegir la mejor configuración para éste. Aunque no es lo habitual, la instalación automatizada de LILO puede fallar, dejando el registro de arranque maestro de su disco inservible (aunque es difícil que ésto llegue a hacerle perder datos de su disco). En concreto, si utiliza el *Boot Manager* de OS/2, usted *no deberá* instalar LILO mediante el procedimiento automático—para ello, habrá que seguir instrucciones especiales que le daremos más tarde.

En muchos casos, lo mejor es usar un disquete de arranque, hasta que usted esté en condiciones de configurar LILO a mano. Si usted es excepcionalmente confiado, puede seguir adelante con el procedimiento automático para instalar LILO si éste forma parte de la distribución.

En el Capítulo 4 comentaremos en detalle cómo configurar e instalar LILO para su configuración particular.

Si todo fue bien, ¡Enhorabuena! Acaba usted de instalar Linux en su sistema. Tómese una Coca-Cola Light o algo así—realmente se lo merece.

En caso de que tuviese algún problema, la siguiente sección le describirá los puntos "oscuros" más habituales en las instalaciones de Linux, y cómo resolverlos.

2.3.8 Otros procedimientos de instalación

Algunas distribuciones proporcionan procedimientos de instalación adicionales, permitiéndole configurar diversos módulos como el de red TCP/IP, el sistema X Window, y otros. Si en su instalación se encontró con opciones de este tipo, puede que desee leer capítulos posteriores de este libro para obtener más información acerca de cómo configurar ese software. Si no, puede guardar esas opciones de instalación hasta que entienda perfectamente cómo configurar el software.

Hasta aquí, si algo falla, retroceda y mire a ver qué puede estar pasando. Es difícil que algo que haya hecho mal ahora no pueda ser deshecho después (toque madera, por si acaso).

2.4 Procedimientos post-instalación

Después de haber completado la instalación de Linux, debería de haber poco más que hacer antes de que Ud. pueda comenzar a usar el sistema. En la mayoría de los casos, Ud. debería poder arrancar el sistema, entrar como `root`, y comenzar a explorar el sistema. (Cada distribución tiene un método ligeramente distinto de hacer esto.)

Llegado este punto es una buena idea explicar cómo rearrancar y apagar el sistema cuando lo esté usando. No debería nunca rearrancar o apagar su sistema Linux presionando el interruptor de reset o con el viejo "Vulcan Never Pinch"—o sea, pulsando a la vez `ctrl-alt-del`.[10] Por supuesto, tampoco debería desconectar la corriente. Como en la mayoría de sistemas UNIX, Linux lleva una *cache* de disco en memoria, lo que aplaza la escritura de los datos. Es por ello que si Ud. rearranca el sistema sin apagarlo "limpiamente", puede corromper datos en sus unidades, causando un daño impredecible.

La forma más fácil de apagar el sistema es usar el comando `shutdown`. Como ejemplo, para apagar y rearrancar el sistema de forma inmediata, use el siguiente comando como `root`:

```
# shutdown -r now
```

Esto apagará limpiamente su sistema. La página de manual de `shutdown` describe el resto de argumentos disponibles en la línea de comandos.[11]

Observe, sin embargo, que muchas distribuciones no proporcionan el comando `shutdown` en el software de instalación. Esto significa que la primera vez que Ud. rearranque su sistema después de la instalación, tendrá que hacer uso de la combinación de teclas `ctrl-alt-del`. Después de esto, deberá usar siempre el comando `shutdown`.

Después de que haya tenido la oportunidad de explorar y usar el sistema, hay varias opciones de configuración que debería revisar. La primera es crear una cuenta de usuario para Ud. mismo (y, opcionalmente, para el resto de usuarios que podrían tener acceso al sistema). La creación de cuentas de usuario se describe en la Sección 4.4. Generalmente, todo lo que Ud. tiene que hacer es entrar como `root`, y ejecutar el programa `adduser` (algunas veces `useradd`). Éste le ayudará por medio de varias preguntas a crear una nueva cuenta de usuario.

Si Ud. creo más de un sistema de archivos para Linux, o si está usando una partición de swap, puede tener que editar el fichero `/etc/fstab` de forma que esos sistemas de archivo puedan estar disponibles después de rearrancar. (Por ejemplo, si está utilizando un sistema de archivos aparte para `/usr`, y ninguno de los ficheros que deberían estar en `/usr` parecen estar presentes, simplemente necesita montar ese sistema de archivos). En la Sección 4.8 se describe este procedimiento. Observe que la distribución Slackware de Linux configura automáticamente sus sistemas de archivo y espacio de swap en tiempo de instalación, por lo que esto no suele ser necesario.

---

[10] En la mayoría de sistemas Linux, sin embargo, `ctrl-alt-del` provocará que el sistema se apague de forma adecuada, como si Ud. hubiera usado el comando `shutdown`.

[11] Use el comando `man shutdown` para ver la página de manual de `shutdown`.

2.5 Resolviendo problemas

Casi todo el mundo se encuentra con algún tipo de pega o cuelgue cuando intenta instalar Linux por primera vez. La mayoría de veces el problema se debe a una simple confusión. Otras veces, sin embargo, puede ser algo más serio, como una equivocación de uno de los desarrolladores, o un error del programa.

Esta sección describirá algunos de los más comunes problemas de instalación, y cómo resolverlos. Si su instalación parece haber tenido éxito, pero recibió mensajes de error inesperados durante la misma, también puede encontrarlos descritos aquí.

2.5.1 Problemas con el arranque

Cuando se intenta arrancar con el *floppy* de arranque la primera vez, se pueden encontrar algunos problemas. Dichos problemas se listan a continuación. Observe que los siguientes *no* están relacionados con el arranque del Linux una vez instalado. Vea la sección 2.5.4 para obtener información acerca de estos problemas.

- **Se produce un error en el *floppy* u otro dispositivo durante el arranque.**

  El motivo más frecuente de esta clase de problemas es que el disquete esté corrupto. Puede ser que el disquete esté físicamente dañado, en cuyo caso tendrá que volverlo a preparar sobre un *nuevo* disquete, o bien que los datos fueran mal copiados al mismo, en cuyo caso debe verificarse si se consiguió la imagen del disquete de arranque correctamente. En muchos casos, basta con volver a grabar la imagen sobre el *floppy*: repita todos los pasos e intente de nuevo.

  Si el disquete de arranque lo recibió de un distribuidor por correo, por ejemplo; y no puede volver a crear el disquete, contacte con el distribuidor para pedirle uno nuevo, pero solo después de comprobar que éste es el problema.

- **El sistema se "cuelga" durante el arranque o después.**

  Después de que el disquete arranque, debe ver una serie de mensajes del núcleo, indicándole qué dispositivos está detectando y configurando. Después de esto, normalmente verá un *prompt* de *login*, que le permite iniciar la instalación (en otras distribuciones se entra directamente en un programa de instalación). El sistema puede parecer "colgado" durante cualquiera de esos pasos. Sea paciente, la carga del disquete es lenta. Muchas veces el sistema no se ha bloqueado, simplemente necesita tiempo. Verifique que no se usa ningún dispositivo del sistema durante algunos minutos antes de estar seguros de que se ha bloqueado la máquina.

  1. Después del *prompt* de `LILO`, el sistema debe cargar el núcleo desde el floppy. Esto puede llevar varios segundos; y puede verse que está sucediendo pues la luz del floppy permanecerá encendida.
  2. Mientras el núcleo arranca, se probarán los dispositivos SCSI. Si su sistema tiene SCSI, el sistema se bloqueará durante unos 15 segundos mientras se prueban esos dispositivos; y sucede normalmente después de verse la línea

     ```
     lp_init: lp1 exists (0), using polling driver
     ```

en su pantalla.

3. Una vez que el núcleo ha terminado de arrancar, se transfiere el control a los ficheros de arranque que hay en el disquete. Finalmente, se verá un *prompt* de entrada en el sistema, o bien se entrará en un programa de instalación. Si se ve un *prompt* como

   ```
   Linux login:
   ```

   deberá entonces entrar (normalmente como `root` o `install`—según la distribución que utilice). Tras esto, el sistema puede tardar aun 20 o más segundos mientras el programa de instalación o el *shell* se carga desde el disquete. De nuevo, la luz del floppy se mantendrá encendida. No crea que el sistema se ha bloqueado.

Cualquier cosa de las comentadas más arriba puede ser la causa de su problema. Sin embargo, es posible que su sistema se "cuelgue" realmente durante el arranque, y eso puede deberse a varias cosas. En primer lugar, puede suceder que no tenga suficiente RAM para arrancar (vea el siguiente punto al respecto o desactive el disco RAM para liberar memoria).

La causa de la mayoría de los "cuelgues" son las incompatibilidades del hardware. En la sección 1.8 del capítulo anterior se presentó una visión general del hardware que soporta Linux. Aunque sus dispositivos estén soportados, puede tener problemas con configuraciones de éstos incompatibles que lleven su sistema al bloqueo. Vea la sección 2.5.2, más adelante, para ver una discusión acerca de estas incompatibilidades.

- **El sistema informa de errores por falta de memoria durante el arranque.**

  Esto tiene que ver con la RAM que tiene. En sistemas con 4 megaoctetos o menos, puede tener problemas al intentar arrancar el disquete, o con los programas de instalación. Esto se debe a que muchas distribuciones utilizan un disco RAM, que es un sistema de ficheros cargado en la memoria, para operaciones durante la instalación. Por ejemplo, toda la imagen del disquete de arranque podría ser cargada en el disco RAM, lo que supone ocupar más de un megaocteto de memoria.

  La solución a este problema es desactivar la opción de disco RAM durante el arranque. Cada versión sigue un procedimiento diferente para hacer esto; en SLS, por ejemplo, se debe teclear "`floppy`" en el *prompt* de LILO cuando se arranca el disco `a1`. Vea los detalles de su distribución en la documentación incluida.

  Puede suceder que en lugar de un mensaje de error por falta de memoria, su sistema se bloquee durante el arranque. Si esto sucede, y no le sirve ninguna recomendación de las vistas en la sección anterior, pruebe a desactivar el disco RAM.

  Tenga en cuenta que Linux requiere por sí mismo un mínimo de 2 megabytes de RAM; y algunas distribuciones necesitan 4 o más.

- **El sistema muestra un error como "`permission denied`" o "`file not found`" durante el arranque.**

  Esto es señal de que su disquete de instalación está mal. Si intenta arrancar con el disquete, y éste es correcto, no deberían salir errores de este tipo. Contacte con su distribuidor de Linux y trate de solucionar el problema, si es necesario obteniendo una nueva copia del disquete de instalación. Si usted creó por su cuenta el disco de arranque, pruebe a rehacerlo para ver si esto soluciona el problema.

- **El sistema informa del error "`VFS: Unable to mount root`" cuando se está arrancando.**

  Este error indica que el sistema de ficheros raíz (que se debe encontrar en el disquete de arranque), no está. Puede ser que su disquete esté mal o que no esté arrancando el sistema de forma correcta.

  Por ejemplo, en muchas distribuciones en CD-ROM, se necesita que tenga el CD en la unidad durante el arranque; además la unidad debe estar encendida y preparada. Es posible también que el sistema no detecte su CD-ROM durante el arranque; para más información, vea la sección 2.5.2.

  Si está seguro que ha seguido correctamente el procedimiento de arranque, puede ser que su disquete esté corrupto. Es poco corriente, por lo que deben buscarse otras soluciones antes que intentar usar otro disquete o cinta.

2.5.2 Problemas con el hardware

El problema más habitual que surge cuando se arranca Linux es la incompatibilidad con el hardware. Aunque todo su hardware esté soportado en Linux, algún conflicto de las configuraciones puede causar extraños resultados—sus dispositivos pueden no detectarse durante el arranque, o el sistema puede bloquearse.

Es importante aislar esos problemas si sospecha que puede ser el origen del mal funcionamiento. En las siguientes secciones describiremos algunos problemas habituales, y cómo resolverlos.

2.5.2.1 Aislando los problemas de hardware

Si se encuentra con un problema que cree que está relaccionado con el hardware, la primera cosa que debe hacer es intentar aislar el problema. Esto significa eliminar todas las variables posibles y (por lo general) ir desmontando, pieza-por-pieza, hasta que la pieza del hardware causante del problema haya sido aislada.

Esto no es tan aterrador como parece. Basicamente, Ud. debería quitar todo el hardware que no sea esencial de su sistema, y entonces determinar que dispositivo esta causando el problema—si es posible reinsertando un dispositivo cada vez. Esto quiere decir que Ud. debería quitar todo el hardware excepto la controladora de discos y la tarjeta de video, y por supuesto el teclado. Incluso aquellos dispositivos que parecen inofensivos, tales como controladores de ratón pueden producirle quebraderos de cabeza a menos que los considere no esenciales.

Por ejemplo, digamos que el sistema se cuelga durante la secuencia de detección de la tarjeta Ethernet al arrancar. Ud. podría suponer que hay un conflicto o problema con la tarjeta Ethernet de su máquina. El modo rápido y sencillo de averiguarlo es quitar la tarjeta Ethernet y arrancar de nuevo. Si todo va bien, entonces ya sabe que (a) Linux no soporta esa tarjeta Ethernet (vea en la Sección 1.8 una lista de tarjetas compatibles), o bien (b) hay un conflicto de dirección o de IRQ con la tarjeta.

"¿Conflicto de dirección o de IRQ?" ¿Qué diablos significa eso? Todos los dispositivos de su máquina usan una *IRQ*, o *Interrupt Request Line (Línea de petición de interrupción)*, para decir al sistema que necesitan que el sistema haga algo por ellos. Puede interpretar una IRQ como una cuerda de la que el dispositivo tira cuando necesita que el sistema se haga cargo de alguna petición pendiente. Si hay más de un dispositivo tirando de la misma cuerda, el núcleo no podrá determinar a que dispositivo debe atender.

Por eso, asegurese de que todos los dispositivos que tenga instalados estén usando líneas IRQ únicas. Por lo general la IRQ de un dispositivo puede establecerse por puentes en la tarjeta; lease la documentación del dispositivo en particular para más detalles. Algunos dispositivos no necesitan del uso de una IRQ en absoluto, pero se sugiere que los configure para usar una si es posible (las controladoras SCSI Seagate ST01 y ST02 son buenos ejemplos).

En algunos casos, el núcleo que le proporciona la instalación está preconfigurado para usar una determinada IRQ para un dispositivo concreto. Por ejemplo, en algunas distribuciones de Linux, el núcleo está preconfigurado para usar la IRQ 5 para la controladora SCSI TMC-950, la controladora de CD-ROM Mitsumi, y el driver de ratón tipo bus. Si Ud. quiere usar dos o más de estos dispositivos, necesitará instalar primero Linux con uno solo de estos dispositivos habilitado, y entonces recompilar el núcleo para cambiar la IRQ por defecto para uno de ellos. (Léase el Capítulo 4 para informarse sobre como recompilar el núcleo.)

Otro área donde pueden surgir conflictos hardware es con los canales DMA (Direct Memory Access - Acceso Directo a Memoria), direcciones E/S, y direcciones de memoria compartida. Todos estos términos describen mecanismos a través de los cuales el sistema interactúa con los dispositivos hardware. Algunas tarjetas Ethernet, por ejemplo, utilizan una dirección de memoria compartida a la vez que una IRQ para interactuar con el sistema. Si alguna de éstas está en conflicto con otros dispositivos, entonces el sistema se puede comportar de modo inesperado. Ud. debería ser capaz de cambiar el canal de DMA, dirección de E/S o memoria compartida en sus diversos dispositivos con configuraciones de puentes (desafortunadamente, algunos dispositivos no permiten cambiar estas configuraciones).

La documentación de sus diversos dispositivos hardware debería especificar la IRQ, canal DMA, dirección E/S, o dirección de memoria compartida que usan los dispositivos, y cómo configurarlos. De nuevo, la manera sencilla de acercarse a estos problemas es deshabilitar temporalmente los dispositivos en conflicto hasta que Ud. tenga tiempo de determinar la causa del problema.

La Tabla 2.2 es una lista de IRQs y canales DMA usados por varios dispositivos "standard" que se encuentran en la mayoría de sistemas. Casi todos los sistemas tendrán alguno de estos dispositivos, así que Ud. debe evitar poner la IRQ o DMA de otros dispositivos en conflicto con estos valores.

2.5.2.2 Problemas reconociendo discos duros o controladoras

Cuando arranca Linux, Ud. debería de ver una serie de mensajes en su pantalla, tales como:

```
Console:  colour EGA+ 80x25, 8 virtual consoles
Serial driver version 3.96 with no serial options enabled
tty00 at 0x03f8 (irq = 4) is a 16450
```

Dispositivo	Dirección E/S	IRQ	DMA
ttyS0 (COM1)	3f8	4	n/a
ttyS1 (COM2)	2f8	3	n/a
ttyS2 (COM3)	3e8	4	n/a
ttyS3 (COM4)	2e8	3	n/a
lp0 (LPT1)	378 - 37f	7	n/a
lp1 (LPT2)	278 - 27f	5	n/a
fd0, fd1 (floppies 1 and 2)	3f0 - 3f7	6	2
fd2, fd3 (floppies 3 and 4)	370 - 377	10	3

Tabla 2.2: Configuraciones comunes de dispositivo

```
tty03 at 0x02e8 (irq = 3) is a 16550A
lp_init:  lp1 exists (0), using polling driver
...
```

En ellos el kernel está detectando los diversos dispositivos hardware presentes en su sistema. En algún momento, debería aparecer la línea

```
Partition check:
```

seguida de una lista de particiones reconocidas, por ejemplo:

```
Partition check:
hda:  hda1 hda2
hdb:  hdb1 hdb2 hdb3
```

Si, por cualquier razón, sus unidades o particiones no son reconocidas, entonces no podrá acceder a ellas de ningún modo.

Hay varias cosas que pueden provocar que esto ocurra:

- **Disco duro o controladora no soportados.** Si Ud. está usando una controladora de disco (IDE, SCSI, o cualquier otra) que no esté soportada por Linux, el kernel no reconocerá sus particiones a la hora de arrancar.
- **Unidad o controladora mal configurada.** A pesar de que su controladora esté soportada por Linux, puede no estar configurada correctamente. (Esto es especialmente problemático con controladoras SCSI; la mayoría de controladoras que no son SCSI deberían funcionar bien sin configuración adicional).

  Consulte la documentación de su disco duro y/o controladora para información sobre como resolver este tipo de problemas. Precisamente, muchos discos duros necesitan tener una determinada combinación de puentes si van a ser usados como discos “esclavo” (por ejemplo, como un segundo disco duro). La manera más sencilla de probar esto es arrancando con MS-DOS, o cualquier otro sistema operativo, que se sepa que funciona con su disco y controladora. Si es posible acceder a la unidad y a la controladora desde otro sistema operativo, entonces no se trata de un problema de configuración del hardware.

Vaya a la Sección 2.5.2.1, anterior, para obtener información sobre cómo resolver posibles conflictos de dispositivos, y la Sección 2.5.2.3, posterior, para información sobre cómo configurar dispositivos SCSI.

- **Controladora bien configurada, pero no detectada.** Algunas controladoras SCSI sin BIOS necesitan que el usuario especifique la información sobre la controladora a la hora de arrancar. La Sección 2.5.2.3, posterior, describe como forzar la detección de hardware para estas controladoras.

- **Geometría del disco duro no reconocida.** Algunos sistemas, como el IBM PS/ValuePoint, no almacenan la información de geometría del disco duro en la memoria CMOS, donde Linux espera encontrarla. También, ciertas controladoras SCSI necesitan que se les diga dónde encontrar la geometría de la unidad de modo que Linux reconozca el diseño de su disco.

  La mayoría de distribuciones proporcionan una opción de arranque para especificar la geometría del disco. En general, cuando se arranca el software de instalación, se puede especificar la geometría del disco en el prompt de arranque de LILO con un comando como:

  `boot:` *linux hd=⟨cylinders⟩,⟨heads⟩,⟨sectors⟩*

  donde ⟨*cylinders*⟩, ⟨*heads*⟩, y ⟨*sectors*⟩ corresponden a el número de cilindros, cabezas y sectores por pista para su disco duro.

  Después de instalar el software de Linux, Ud. podrá instalar LILO, permitiendole arrancar desde el disco duro. En ese momento, puede especificar la geometría del disco al procedimiento de instalación de LILO, haciendo innecesario que tenga que introducirlo cada vez que arranque. Lease el Capítulo 4 para mayor información sobre LILO.

2.5.2.3 Problemas con controladoras y unidades SCSI

Aquí tiene algunos de los problemas más comunes con controladoras y unidades SCSI, tales como CD-ROMs, discos duros, y unidades de cinta. Si está teniendo problemas haciendo que Linux reconozca su unidad o controladora, siga leyendo.

El HOWTO SCSI de Linux (ver Apéndice A) contiene mucha información útil sobre unidades SCSI además de la aquí listada. A veces SCSI puede ser particularmente exigente de configurar.

- **Una unidad SCSI se detecta en todos los ID's posibles.** La causa de esto es que la unidad esta ligada a la misma dirección que la controladora. Es necesario cambiar la configuración de los jumpers de modo que la unidad use una dirección distinta de la de la propia controladora.

- **Linux reporta errores de detección, incluso sabiendo que las unidades están libres de errores.** Esto puede estar provocado por cables en mal estado, o malos terminadores. Si su bus SCSI no esta terminado en ambos extremos, Ud. puede tener errores accediendo a las unidades SCSI. Cuando tenga dudas, compruebe sus cables.

- **Las unidades SCSI dan errores de timeout.** Eso se produce generalmente por un conflicto con las IRQ, DMA o direcciones de unidad. Compruebe también que las interrupciones estén correctamente activadas en su controladora.

- **Las controladoras SCSI que utilizan BIOS no son detectadas.** La detección de controladoras que usan BIOS fallará si la BIOS está desactivada, o si la "firma" de su controladora no es reconocida por el kernel. Lease el Linux SCSI HOWTO para mayor información sobre esto.

- **Las controladoras que utilizan E/S mapeada en memoria no funcionan.** Esto sucede cuando los puertos de E/S mapeados en memoria están incorrectamente cacheados. Puede marcar el espacio de direccionamiento de la tarjeta como no cacheable en las opciones de XCMOS, o bien deshabilite la cache totalmente.

- **Cuando particiona, se obtiene un aviso de "cylinders > 1024", o le es imposible arrancar desde una partición que utilice cilindros numerados por encima de 1023.** La BIOS limita el número de cilindros a 1024, y cualquier partición que use cilindros numerados por encima de éstos no será accesible desde la BIOS. Por lo que respecta al Linux, esto solo afecta al arranque; una vez que el sistema ha arrancado, debería poder acceder a la partición. Sus opciones son arrancar, bien desde un disquete de arranque, o bien desde una partición que utilice los cilindros por debajo del 1024. Lea la Sección 2.3.7 para obtener información sobre cómo crear un disquete de arranque o instalar LILO.

- **La unidad CD-ROM u otras unidades de información removible no se reconocen a la hora de arrancar.** Pruebe a arrancar con un CD-ROM (o disco) en la unidad. Esto es necesario en algunos dispositivos.

Si no se reconoce su controladora SCSI, puede que tenga que forzar la detección del hardware en el momento del arranque. Esto es especialmente importante para controladoras SCSI sin BIOS. La mayoría de las distribuciones le permiten especificar la IRQ de la controladora y la dirección de memoria compartida cuando arranca con los discos de instalación. Por ejemplo, si Ud. está usando una controladora TMC-8xx, deberá poder introducir

`boot:` *linux tmx8xx=⟨interrupción⟩,⟨dirección-de-memoria⟩*

en el prompt de arranque de LILO, donde ⟨*interrupción*⟩ es la IRQ de la controladora, y ⟨*dirección-de-memoria*⟩ es la dirección de memoria compartida. El que Ud. pueda o no hacer esto depende de la distribución de Linux que esté usando; consulte su documentación para más detalle.

2.5.3 Problemas con la instalación del software

Con un poco de suerte, se puede instalar el software de Linux sin problemas. Los únicos que suelen aparecer se relacionan con los errores en los disquetes de instalación o con el espacio disponible en los sistemas de ficheros. A continuación se relaciona una lista de estos problemas.

- **El sistema muestra errores como "`Read error`", "`file not found`" durante la instalación del software.** Esto es indicativo de problemas en los disquetes o cintas de instalación. Si se instala desde disquetes, tenga en cuenta que los errores en éstos son posibles. Asegúrese de que está utilizando disquetes nuevos o recién formateados. Muchas distribuciones permiten instalar el software desde una partición DOS del disco duro. Esto puede ser más seguro y más rápido que usar directamente los disquetes.

  Si utiliza un CD-ROM, asegúrese de que el disco no tiene rayaduras o suciedad que pudieran ser causa de errores de lectura.

  La causa del problema puede estar también en un formato incorrecto de los disquetes. Normalmente se exige que los disquetes estén en formato MS-DOS de alta densidad (a excepción del disquete de arranque, que suele tener su propio formato casi siempre). Si todo esto falla, intente obtener nuevos disquetes, bien sea pidiéndolos a su distribuidor o construyéndolos usted mismo.

- **El sistema da errores tipo "`tar: read error`" o "`gzip: not in gzip format`".**

  Este problema suele deberse a errores en los ficheros o en los propios discos o cintas. En otras palabras, sus disquetes pueden no tener errores, pero sí los datos contenidos en ellos. Por ejemplo, un error común es obtener los ficheros por las redes con modo de transferencia ASCII en lugar de binario, lo que hace inservibles los ficheros obtenidos.

- **El sistema da errores como "`device full`" durante la instalación.**

  Esto es un signo claro de que se está intentando instalar Linux sin espacio de disco suficiente. En la mayoría de las distribuciones, no puede esperarse que el sistema funcione abortando el proceso de instalación.

  La solución habitual es rehacer los sistemas de ficheros (mediante el comando `mke2fs`) lo que borrará el software parcialmente instalado. Ahora puede reintentar la instalación, seleccionando menos componentes para instalar. En otros casos, puede necesitarse comenzar desde cero, rehaciendo particiones y sistemas de ficheros.

- **El sistema informa de errores como "`read_intr: 0x10`" durante los accesos al disco duro.**

  Esto suele deberse a la presencia de bloques con errores en el disco. Sin embargo, si se reciben estos errores al utilizar `mkswap` o `mke2fs`, el sistema puede estar teniendo problemas para acceder a su controlador. Puede ser tanto un problema del hardware (vea la sección 2.5.2) o una incorrecta especificación de la geometría del disco. Si utilizó la opción

  `hd=`⟨*cylinders*⟩`,`⟨*heads*⟩`,`⟨*sectors*⟩

  en el momento de arrancar para especificar la geometría de su disco, y lo hizo incorrectamente, puede encontrarse con estos problemas. También puede suceder si la información de la CMOS acerca de la geometría del disco no es correcta.

- **El sistema da errores como "`file not found`" o "`permission denied`".**

  Este problema puede suceder si no están disponibles todos los ficheros necesarios en los disquetes de instalación (vea el párrafo siguiente) o si hay problemas con los permisos sobre dichos ficheros. Por ejemplo, en algunas distribuciones de Linux existen bugs rápidamente solucionados en posteriores versiones, relacionados con los permisos. Son errores poco frecuentes. Si sospecha que su distribución tiene bugs, y está seguro de no haber hecho nada mal, contacte con el fabricante de la distribución para informarle del bug.

Si usted tiene otros extraños problemas durante la instalación de Linux (especialmente si el software lo ha obtenido vía red o módem), asegúrese de haber obtenido todos los ficheros necesarios. Por ejemplo, hay gente que utiliza el comando de FTP

```
mget *.*
```

cuando obtiene el software vía FTP. En realidad, este comando solo obtiene todos los ficheros que contengan un "`.`" en el nombre, y no todos lo tienen. El comando correcto a utilizar será

```
mget *
```

La mejor opción es retroceder sobre sus pasos cuando algo vaya mal. Puede creer haber hecho todo correctamente, cuando en realidad olvidó algún paso de apariencia simple pero importante. En muchos casos, bastará con volver a instalar (u obtener) el software para resolver el problema.

También, si Linux bloquea su ordenador durante la instalación de forma inesperada, puede haber algún problema con el hardware. Vea la sección 2.5.2 para más detalle.

2.5.4 Problemas después de instalar Linux

Usted se ha pasado una tarde instalando Linux. Con el fin de dejar espacio para éste, redujo las particiones de MS-DOS u OS/2 y borró con lagrimas en los ojos sus copias de SimCity y Wing Commander. Después arrancó el sistema, y no pasó nada. O, por el contrario, sí pasó *algo*, pero no lo que debería pasar. ¿Qué hace usted ahora?

En la sección 2.5.1 hemos cubierto los problemas que pueden surgir cuando se arranca Linux desde los disquetes de la distribución—aquí suelen concentrarse la mayoría de los problemas. Además, usted puede ser víctima de alguno de los siguientes problemas.

2.5.4.1 Problemas al arrancar Linux desde el floppy

Si utiliza un disquete para arrancar Linux, puede ser que necesite indicar cuál es su partición raíz de Linux en el momento de arrancar. Esto es especialmente cierto si utiliza el disquete de instalación original, y no un disquete personalizado durante la instalación.

Mientras arranca con el disquete, mantenga pulsadas las teclas `shift` o `ctrl`. Esto deberá presentarle un menú de arranque; pulse la tecla `tab` para ver una lista de opciones disponibles. Por ejemplo, en muchas distribuciones se le permitirá teclear

`boot:` *linux root=⟨partition⟩*

en el menú de arranque, donde ⟨*partition*⟩ es el nombre de la partición raíz de Linux, como `/dev/hda2`. Consulte la documentación de su distribución para más detalles.

2.5.4.2 Problemas al arrancar Linux desde el disco duro

Si optó por instalar LILO, en lugar de crear un disquete de arranque, debe poderse arrancar Linux desde el disco duro. Sin embargo, el procedimiento automático de instalación de LILO que muchas distribuciones tienen no siempre es perfecto. Puede tener información incorrecta acerca de su esquema de particiones, en cuyo caso puede que tenga que reinstalar LILO para dejarlo todo correcto. La instalación de LILO se trata en el capítulo 4.

- **El sistema da el mensaje "`Drive not bootable---Please insert system disk.`"** Se obtiene este error cuando el registro maestro de arranque del disco duro (MBR) está mal por alguna causa. Normalmente, el resto de la información del disco estará intacta. Puede entonces suceder:

  1. Mientras se hacen las particiones mediante `fdisk`, puede haberse borrado la partición marcada como "activa". MS-DOS y otros sistemas operativos intentan arrancar desde la partición "activa" (esto a Linux le da igual). Puede entonces arrancar MS-DOS desde un disquete y ejecutar el programa `FDISK` para poner el flag de activo a su partición de MS-DOS.

     Otro comando que se puede intentar (con versiones de MS-DOS iguales o superiores a la 5.0) es

     ```
     FDISK /MBR
     ```

     Este comando intentará reconstruir el registro maestro de arranque del disco (MBR) para arrancar MS-DOS, borrando a LILO. Si no va a tener MS-DOS en su disco duro, necesitará arrancar después Linux desde un disquete e intentar instalar LILO de nuevo.

  2. Si creó particiones para MS-DOS utilizando la versión de `fdisk` para Linux, puede obtenerse este error. Las particiones de MS-DOS solo deben crearse utilizando el comando `FDISK` de MS-DOS. (Esto afecta también a otros sistemas operativos.) La mejor solución es empezar desde el principio y reparticionar los discos correctamente, o simplemente borrar y rehacer particiones utilizando la versión apropiada de `fdisk`.

  3. El procedimiento de instalación de LILO puede no haber ido bien. En este caso, debe arrancarse Linux desde un disquete (si lo tiene) o desde el medio de arranque original. En cualquiera de los dos casos deberían proporcionarse opciones para especificar la partición raíz de Linux para arrancar. Mantenga pulsada la tecla `shift` o `ctrl` durante el arranque y pulse `tab` en el menú de arranque para ver las opciones.

- **Cuando se arranca desde el disco duro, MS-DOS (u otro) arranca en lugar de hacerlo Linux.** En primer lugar, asegúrese de que instaló LILO mientras instalaba el software de Linux. Si no, el sistema arrancará MS-DOS (u otro) cuando intente arrancar desde el disco duro. Para arrancar Linux desde el disco duro, tiene que instalar LILO (vea el capítulo 4).

  Por otro lado, si usted *instaló* LILO, pero el sistema operativo que arranca automáticamente es otro que no sea Linux, es porque lo ha configurado para arrancar otro sistema operativo por defecto. Mientras arranca el sistema, puede pulsar las teclas `shift` o `ctrl`, y luego `tab` en el *prompt* de arranque para ver una lista de posibles sistemas operativos para arrancar; seleccione la opción apropiada (normalmente etiquetada con "`Linux`") para arrancar este sistema operativo.

Si desea que Linux sea el sistema que entra por defecto, necesitará reinstalar LILO. Vea el capítulo 4.

También es posible que intentara instalar LILO, pero fallara la instalación por algún motivo. Repase lo visto anteriormente.

2.5.4.3 Problemas al entrar en Linux

Después de arrancar Linux, debe verse un *prompt* de *login*, como:

```
linux login:
```

En este punto, la documentación de su distribución le dirá lo que hay que hacer. En muchos casos, hay que entrar como `root` sin password. Otros posibles nombres de usuario iniciales son `guest` o `test`.

Casi siempre no se requieren passwords en los *logins* iniciales. Sin embargo, si se le pide password, puede ser un problema. Primero, pruebe a introducir como password el mismo nombre del usuario (root, guest...).

Si simplemente no puede entrar, consulte la documentación de su distribución. El nombre de usuario y password a utilizar puede estar escondido ahí. También puede habérsele dado durante la instalación o puede estar delante justo del prompt de login.

Una causa de esto es una incorrecta instalación de los ficheros de inicialización del sistema. Si este es el caso, habrá que reinstalar (al menos parte de) el software de Linux, o arrancar desde un disquete de instalación y resolver el problema a mano—vea el capítulo 4 para más detalles.

2.5.4.4 Problemas utilizando el sistema

Si consigue entrar en el sistema, deberá ver un *prompt* de *shell* (como "`#`" o "`$`") y podrá navegar felizmente por su sistema. Sin embargo, existen todavía algunos posibles problemas.

El más habitual sucede con los permisos sobre ficheros o directorios, que puede originar un mensaje de error como

```
Shell-init:  permission denied
```

tras entrar en el sistema (también, durante su sesión en el mismo puede ver el mensaje "`permission denied`". En cualquier caso indica que hay problemas con los permisos en ficheros o directorios).

En muchos casos, basta con utilizar el comando `chmod` para corregir los permisos de los ficheros y directorios. Por ejemplo, en algunas distribuciones de Linux se pone (incorrectamente) el modo 0644 al directorio raíz (`/`). La corrección sería teclear el comando

`#` *chmod 755 /*

desde `root`. Sin embargo, para poder ejecutar este comando, puede necesitar arrancar desde el floppy y montar el sistema de ficheros raíz a mano, lo que puede ser una difícil tarea para los recién llegados.

Conforme utilice el sistema, se encontrará lugares donde los permisos puestos a ficheros y directorios son incorrectos, o el software no trabaja como se esperaba. ¡Bienvenido entonces al mundo de Linux! Mientras que la mayor parte de las distribuciones casi no dan problemas, lo cierto es que muy pocas son perfectas. No queremos cubrir todos esos problemas aquí. En su lugar, a lo largo del libro le ayudaremos a resolver muchos de estos problemas enseñándole cómo encontrar y corregir por sí mismo los errores. En el capítulo 1 ya comentamos en parte esta filosofía. En el capítulo 4, le daremos consejos para corregir muchos de esos problemas con la configuración.

Capítulo 3

Tutorial de Linux

3.1 Introducción

Los nuevos usuarios de UNIX y Linux pueden estar un poco intimidados por el tamaño y aparente complejidad del sistema que tienen ante si. Hay muchos buenos libros sobre el uso de UNIX para todos los niveles, desde novatos a expertos. Pero ninguno de estos libros cubre especificamente una introducción al uso de Linux. Mientras el 95% del uso de Linux es exactamente como cualquier otro UNIX, la forma más clara de comenzar con su nuevo sistema es un tutorial a medida para Linux. He aquí ese tutorial.

Este capítulo no presentará gran cantidad de detalles o cubrirá temas muy avanzados. Sino que está pensado para permitir al nuevo usuario de Linux comenzar a usar el sistema y situarlo en una posición en la que él o ella puedan leer libros más generales sobre UNIX y entender las diferencias básicas entre otros sistemas UNIX y Linux.

Se va a presuponer muy poco, excepto quizá alguna familiaridad con los ordenadores personales y MS-DOS. Pero incluso si no es un usuario de MS-DOS, debería ser capaz de entender cualquier cosa de las que hablemos. A primera vista, UNIX parece como MS-DOS (después de todo, partes de MS-DOS fueron tomadas de CP/M, el cual fué a su vez inspirado en UNIX). Pero, solo las características superficiales de UNIX se parecen a MS-DOS. Incluso si es completamente nuevo en el mundo de los PC, este tutorial debería serle de ayuda.

Y, antes de comenzar: *No tenga miedo de experimentar.* El sistema no le morderá. No puede destruir nada trabajando con el sistema. UNIX tiene ciertos sistemas de seguridad para evitar que usuarios 'normales' (del tipo que suponemos que es usted) dañen ficheros esenciales para el sistema. Incluso si ocurre el peor de los casos que es que borre todos sus ficheros, tendrá que volver atrás y reinstalar el sistema, pero incluso en ese caso, no hay nada que perder.

3.2 Conceptos básicos de UNIX

UNIX es un sistema operativo multitarea y multiusuario. Esto significa que puede haber más de una persona usando un ordenador a la vez, cada uno de ellos ejecutando a su vez diferentes aplicaciones. (Esto difiere de MS-DOS, donde solo una persona puede usar el sistema en un momento dado). Bajo UNIX, para que los usuarios puedan identificarse en el sistema, deben **presentarse (log in)**, proceso que consta de dos pasos: Introducir el **nombre de usuario (login)** (el nombre con que será identificado por el sistema), y una **palabra de paso (password)**, la cual es su llave personal secreta para entrar en la cuenta. Como solo usted conoce su palabra de paso, nadie más podrá presentarse en el sistema con su nombre de usuario.

En los sistemas UNIX tradicionales, el administrador del sistema asignará el nombre de usuario y una palabra de paso inicial en el momento de crear la cuenta de usuario. Como usted es el administrador del sistema, debe configurar su propia cuenta antes de poder presentarse—ver Sección 3.2.1 más adelante. Para el resto de las discusiones, usaremos el nombre de usuario "`larry`".

Además, cada sistema UNIX tiene un **nombre del sistema (hostname)** asignado. Este "hostname" le da nombre a la máquina, además de carácter y encanto. El nombre del sistema es usado para identificar máquinas en una red, pero incluso aunque la máquina no esté en red, debería tener su nombre. En la Sección 4.10.2 veremos como inicializar el nombre de la máquina. En nuestros ejemplos, el nombre del sistema será "`mousehouse`"

3.2.1 Creación de una cuenta

Antes de poder usar el sistema, deberá configurarse una cuenta de usuario. Esto es necesario, porque no es buena idea usar la cuenta de `root` para los usos normales. La cuenta de `root` debería reservarse para el uso de comandos privilegiados y para el mantenimiento del sistema, como se verá en la Sección 4.1.

Para crear su propia cuenta, necesita entrar en la cuenta de `root` y usar las órdenes `useradd` o `adduser`. Ver la Sección 4.4 para información sobre este procedimiento.

3.2.2 Presentación en el sistema (loggin in)

En el momento de presentarse en el sistema, verá la siguiente línea de comandos en la pantalla:

```
mousehouse login:
```

Ahora, introduzca su nombre de usuario y pulse Return. Nuestro heroe `larry`, teclearía lo siguiente:

```
mousehouse login: larry
Password:
```

Ahora introduzca la palabra de paso. Esta no será mostrada en la pantalla conforme se va tecleeando, por lo que debe teclear cuidadosamente. Si introduce una palabra de paso incorrecta, se mostrará el siguiente mensaje

```
Login incorrect
```

y deberá intentarlo de nuevo.

Una vez que ha introducido correctamente el nombre de usuario y la palabra de paso, está oficialmente "presentado" en el sistema y libre para comenzar a trabajar.

3.2.3 Consolas virtuales

La **consola** del sistema es el monitor y teclado conectado directamente al sistema. (Como UNIX es un sistema operativo multiusuario, puede tener otros terminales conectados a puertos serie del sistema, pero estos no serán la consola). Linux, como otras versiones de UNIX, porporciona acceso a **consolas virtuales** (o VC's), las cuales le permitirán tener más de una sesión de trabajo activa desde la consola a la vez.

Para demostrar esto, entre en su sistema (como hemos visto antes). Ahora pulse `alt-F2`. Debería ver la pregunta `login:` de nuevo. Está viendo la segunda consola virtual—ha entrado en el sistema por la primera. Para volver a la primera VC, pulse `alt-F1`. *Voila!* ha vuelto a la primera sesión.

Un sistema Linux recién instalado probablemente le permita acceder a las primeras cuatro VC's, usando `alt-F1` a `alt-F4`. Pero es posible habilitar hasta 12 VC's— una por cada tecla de función del teclado. Como puede ver, el uso de VC's es muy potente— puede estar trabajando en diferentes VC's a la vez.

Mientras que el uso de VC's es algo limitado (después de todo, solo puede mirar un VC cada vez), esto debería darle una idea de las capacidades multiusuario del sistema. Mientras está trabajando en el VC #1, puede conmutar al VC #2 y comenzar a trabajar en otra cosa.

3.2.4 Intérpretes de comandos y comandos

En la mayoría de las exploraciones en el mundo de UNIX, estará hablando con el sistema a través del uso de un **intérprete de comandos**. Un intérprete de comandos es simplemente un programa que toma la entrada del usuario (p.ej. las órdenes que teclea) y las traduce a instrucciones. Esto puede ser comparado con el `COMMAND.COM` de MS-DOS, el cual efectua esencialmente las misma tarea. El intérprete de comandos es solo uno de los interfaces con UNIX. Hay muchos interfaces posibles— como el sistema X Windows, el cual le permite ejecutar comandos usando el ratón y el teclado.

Tan pronto como entra en el sistema, el sistema arranca un intérprete de comandos y Ud. ya puede teclear órdenes al sistema. Veamos un ejemplo rápido. Aquí, Larry entra en el sistema y es situado en el intérprete de comandos

```
mousehouse login:  larry
Password:  larry's password
Welcome to Mousehouse!

/home/larry#
```

"`/home/larry#`" es el "prompt" del intérprete de comandos, indicando que está listo para recibir órdenes. Tratemos de decirle al sistema que haga algo interesante:

```
/home/larry# make love
make:  *** No way to make target 'love'.  Stop.
/home/larry#
```

Bien, como resulta que `make` es el nombre de un programa ya existente en el sistema, el intérprete de comandos lo ejecuta. (Desafortunadamente, el sistema no está siendo muy amigable).

Esto nos lleva a una cuestión importante: ¿Qué son órdenes? ¿Qué ocurre cuando tecleamos "`make love`"? La primera palabra de la orden, "`make`", es el nombre de la orden a ejecutar. El resto de la orden es tomado como argumentos de la orden. Ejemplos:

```
/home/larry# cp foo bar
```

Aquí, el nombre de la orden es "`cp`", y los argumentos son "`foo`" y "`bar`".

Cuando teclea una orden, el intérprete de comandos hace varias cosas. Primero de todo, busca el nombre de la orden y comprueba si es una orden interna. (Es decir, una orden que el propio intérprete de comandos sabe ejecutar por si mismo. Hay bastantes órdenes de ese tipo que veremos más adelante). El intérprete de comandos también comprueba si la orden es un "alias" o nombre sustitutorio de otra orden. Si no se cumple ninguno de estos casos, el intérprete de comandos busca el programa y lo ejecuta pasándole los argumentos especificados en la línea de comandos.

En nuestro ejemplo, el intérprete de comandos busca el programa llamado `make` y lo ejecuta con el argumento `love`. `make` es un programa usado a menudo para compilar programas grandes, y toma como argumentos el nombre de un "objetivo" a compilar. En el caso de "`make love`", ordenamos a `make` que compile el objetivo `love`. Como `make` no puede encontrar un objetivo de ese nombre, falla enviando un mensaje de error y volviendo al intérprete de comandos.

¿Qué ocurre si tecleamos una orden y el intérprete de comandos no puede encontrar el programa de ese nombre? Bien, probémoslo:

```
/home/larry# eat dirt
eat:  command not found
/home/larry#
```

Bastante simple, si no se puede encontrar el programa con el nombre dado en la orden (aquí "`eat`"), se muestra un mensaje de error que debería de ser autoexplicativo. A menudo verá este mensaje de error si se equivoca al teclear una orden (por ejemplo, si huviese tecleado "`mkae love`" en lugar de "`make love`".

3.2.5 Salida del sistema

Antes de ahondar más, deberíamos ver como salir del sistema. Desde la línea de órdenes usaremos la orden

```
/home/larry# exit
```

para salir. Hay otras formas, pero esta es la más fácil.

3.2.6 Cambiando la palabra de paso

También debe asegurarse de la forma de cambiar su palabra de paso. La orden `passwd` le pedirá su palabra de paso vieja y la nueva. Volverá a pedir una segunda vez la nueva para validarla. Tenga cuidado de no olvidar su palabra de paso— si eso ocurre, deberá pedirle al administrador del sistema que la modifique por usted. (Si es el administrador del sistema, vea la Sección 4.4.)

3.2.7 Ficheros y directorios

Bajo la mayoría de los sistemas operativos (UNIX incluido), existe el concepto de **fichero**, el cual es un conjunto de información al que se le ha asignado un nombre (llamado **nombre del fichero**). Ejemplos de fichero son un mensaje de correo, o un programa que puede ser ejecutado. Esencialmente, cualquier cosa salvada en el disco es guardada en un fichero individual.

Los ficheros son identificados por sus nombres. Por ejemplo, el fichero que contiene su historial podría ser salvado con el nombre `history-paper`. Estos nombres usualmente identifican el fichero y su contenido de alguna forma significativa para usted. No hay un formato estándar para los nombres de los ficheros como lo hay en MS-DOS y en otros sistemas operativos; en general estos pueden contener cualquier carácter (excepto `/` — ver la discusión sobre “pathnames” (rutas de ficheros) más adelante), y están limitados a 256 caracteres de longitud.

Con el concepto de fichero aparece el concepto de directorio. Un **directorio** es simplemente una colección de ficheros. Puede ser considerado como una “carpeta” que contiene muchos ficheros diferentes. Los directorios también tienen nombre con el que los podemos identificar. Además, los directorios mantienen una estructura de árbol; es decir, directorios pueden contener otros directorios.

Un fichero puede ser referenciado por su **nombre con camino**, el cual está constituido por su nombre, antecedido por el nombre del directorio que lo contiene. Por ejemplo, supongamos que Larry tiene un directorio de nombre `papers` que contiene tres ficheros: `history-final`, `english-lit` y `masters-thesis`. (Cada uno de los tres ficheros contiene información sobre tres de los proyectos en los que Larry está trabajando). Para referirse al fichero `english-lit`, Larry puede especificar su camino:

```
papers/english-lit
```

Como puede ver, el directorio y el nombre del fichero van separados por un carácter `/`. Por esta razón, los nombres de fichero no pueden contener este car’acter. Los usuarios de MS-DOS encontrarán esta convención familiar, aunque en el mundo MS-DOS se usa el carácter `\`).

Como hemos mencionado, los directorios pueden anidarse uno dentro de otro. Por ejemplo, supongamos que Larry tiene otro directorio dentro de `papers` llamado `cheat-sheet`. El camino de este fichero sería

```
papers/notes/cheat-sheet
```

Por lo tanto, el camino realmente es la "ruta" que se debe tomar para localizar a un fichero. El directorio sobre un subdirectorio dado es conocido como el **directorio padre**. Aquí, el directorio `papers` es el padre del directorio `notes`.

3.2.8 El árbol de directorios

La mayoría de los sistemas UNIX tienen una distribución de ficheros estándar, de forma que recursos y ficheros puedan ser fácilmente localizados. Esta distribución forma el árbol de directorios, el cual comienza en el directorio "`/`", también conocido como "directorio raíz". Directamente por debajo de `/` hay algunos subdirectorios importantes: `/bin`, `/etc`, `/dev` y `/usr`, entre otros. Estos a su vez contienen otros directorios con ficheros de configuración del sistema, programas, etc.

En particular, cada usuario tiene un **directorio "home"**. Este es el directorio en el que el usuario guardará sus ficheros. En los ejemplos anteriores, todos los ficheros de Larry (como `cheat-sheer` y `history-final`) estaban contenidos en el directorio home de Larry. Usualmente, los directorios home de los usuarios cuelgan de `/home` y son nombrados con el nombre del usuario al que pertenecen. Por lo tanto, el directorio "home" de Larry es `/home/larry`.

En la figura 3.1 se muestra un árbol de directorio de ejemplo. Este debería darle una idea de como está organizado en su sistema el árbol de directorios.

3.2.9 Directorio de trabajo actual

En cualquier momento, las órdenes que teclee al intérprete de comandos son dadas en términos de su **directorio de trabajo actual**. Puede pensar en su directorio actual de trabajo como en el directorio en el que actualmente está "situado". Cuando entra en el sistema, su directorio de trabajo se inicializa a su directorio home—`/home/larry` en nuestro caso. En cualquier momento que referencie a un fichero, puede hacerlo en relación a su directorio de trabajo actual, en lugar de especificar el camino completo del fichero.

Vemos un ejemplo. Larry tiene el directorio `papers`, y `papers` contiene el fichero `history-final`. Si Larry quiere echar un vistazo a ese fichero, puede usar la orden

```
/home/larry# more /home/larry/papers/history-final
```

La orden `more` simplemente muestra el fichero, pantalla a pantalla. Pero, como el directorio de trabajo actual de Larry es `/home/larry`, podría haberse referido al fichero de forma *relativa* a su directorio de trabajo actual. La orden sería

```
/home/larry# more papers/history-final
```

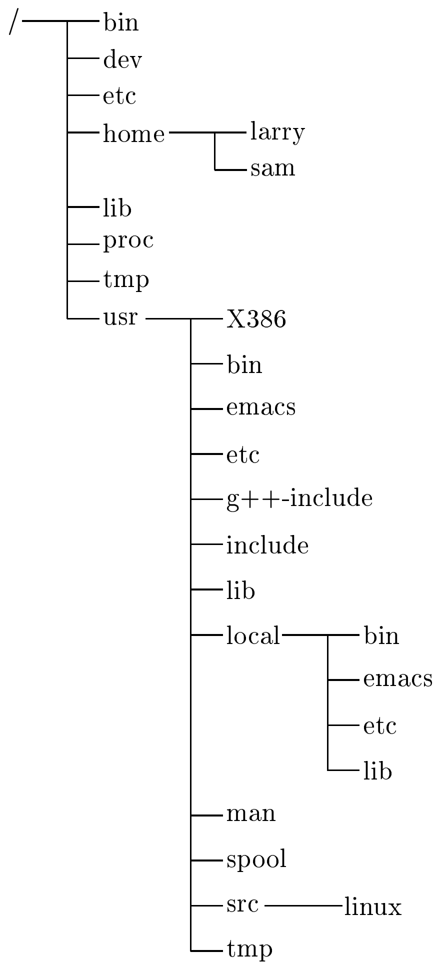

Figura 3.1: Típico árbol de directorios Unix (resumido).

Por lo tanto, si comienza el nombre de un fichero (como `papers/final`) con un carácter distinto a "`/`", el sistema supone que se está refiriendo al fichero con su posición relativa a su directorio de trabajo. Esto es conocido como **camino relativo**.

Por otra parte, si comienza el nombre del fichero con "`/`", el sistema interpreta esto como un camino completo—es decir, el camino al fichero completo desde el directorio raíz, `/`. Esto es conocido como **camino absoluto**.

3.2.10 Refiriendose al directorio home

Bajo `tcsh` y `bash`,[1] el directorio "home" puede ser referenciado usando el carácter de la tilde (~). Por ejemplo, la orden

`/home/larry#` *more ~/papers/history-final*

es equivalente a

`/home/larry#` *more /home/larry/papers/history-final*

El carácter "~" es simplemente sustituido por el intérprete de comandos, con el nombre del directorio home.

Además, también puede especificar otros directorios home de usuarios con la tilde. El camino "`~karl/letters`" es traducido por el intérprete de órdenes a "`/home/karl/letters`" (si `/home/karl` es el directorio home de karl). El uso de la tilde es simplemente un atajo; no existe ningún directorio llamado "~"—es simplemente una ayuda sintáctica proporcionada por el intérprete de comandos.

3.3 Primeros pasos en UNIX

Antes de comenzar es importante destacar que todos los nombres de ficheros y comandos son "case-sensitive" (que hacen diferencia entre mayúsculas y minúsculas, a diferencia de sistemas operativos como MS-DOS). Por ejemplo, el comando `make` es diferente a `Make` o `MAKE`. Lo mismo ocurre en el caso de nombres de ficheros o directorios.

3.3.1 Moviendonos por el entorno

Ahora que ya podemos presentarnos como usuarios, y sabemos como indicar ficheros con su camino completo, ¿cómo podemos cambiar nuestro directorio de trabajo?

La orden para movernos por la estructura de directorios es `cd`, abreviación de "cambio de directorio". Hay que destacar, que la mayoría de las órdenes Unix más usadas son de dos o tres letras. La forma de uso de la orden `cd` es:

`cd` ⟨*directorio*⟩

donde ⟨*directorio*⟩ es el nombre del directorio al que queremos ir.

Como dijimos, al entrar al sistema comenzamos en el directorio "home". Si Larry quiere ir al subdirectorio `papers`, debería usar la orden

[1] `tcsh` y `bash` son dos *intérpretes de comandos* que corren bajo Linux. Un intérprete de comandos es el programa que lee las órdenes del usuario y las ejecuta; la mayoría de los sistemas Linux habilitan `tcsh` o `bash` para las nuevas cuentas de usuario.

```
/home/larry# cd papers
/home/larry/papers#
```

Como se puede ver, la línea de comandos de Larry cambia para mostrar su directorio actual de trabajo. Ahora que ya está en el directorio `papers` puede echarle un vistazo a su fichero history-final con el comando

```
/home/larry/papers# more history-final
```

Ahora Larry está en el subdirectorio `papers`, para volver al directorio padre de este, usará la orden

```
/home/larry/papers# cd ..
/home/larry#
```

(Dese cuenta del espacio entre "`cd`" y "`..`"). Cada directorio tiene una entrada de nombre "`..`" la cual se refiere al directorio padre. De igual forma, existe en cada directorio la entrada "`.`" la cual se refiere a si mismo. Así que el comando

```
/home/larry/papers# cd .
/home/larry#
```

nos deja donde estamos.

También pueden usarse nombres con el camino absoluto en la orden `cd`. Para ir al directorio de Karl con `cd`, introduciremos la siguiente orden.

```
/home/larry/papers# cd /home/karl
/home/karl#
```

También, usando `cd` sin argumentos nos llevará a nuestro directorio de origen.

```
/home/karl# cd
/home/larry#
```

3.3.2 Mirando el contenido de los directorios

Ahora que ya sabe como moverse por los directorios probablemente pensará: ¿Y bien? El simple movimiento por el árbol de directorios es poco útil, necesitamos un nuevo comando, `ls`. `ls` muestra por el terminal la lista de ficheros y directorios, por defecto, los del directorio activo. Por ejemplo;

```
/home/larry# ls
Mail
letters
papers
/home/larry#
```

Aquí podemos ver que Larry tiene tres entradas en su directorio actual: `Mail`, `letters` y `papers`. Esto no nos dice demasiado—¿son ficheros o directorios?. Podemos usar la opción `-F` de la orden `ls` para obtener más información.

```
/home/larry# ls -F
Mail/
letters/
papers/
/home/larry#
```

Por el carácter `/` añadido a cada nombre sabemos que las tres entradas son subdirectorios.

La orden `ls -F` puede también añadir al final "`*`", esto indica que es un fichero **ejecutable**. Si `ls -F` no añade nada, entonces es un fichero normal, es decir no es ni un directorio ni un ejecutable.

Por lo general cada orden UNIX puede tomar una serie de opciones definidas en forma de argumentos. Estos usualmente comienzan con el carácter "`-`", como vimos antes con `ls -F`. La opción `-F` le dice a `ls` que de más información sobre el tipo de ficheros—en este caso añadiendo un `/` detrás de cada nombre de un directorio.

Si a `ls` le pasamos un nombre de directorio, mostrará el contenido de ese directorio.

```
/home/larry# ls -F papers
english-lit
history-final
masters-thesis
notes/
/home/larry#
```

Para ver un listado más interesante, veamos el contenido de directorio del sistema `/etc`.

```
/home/larry# ls /etc

Images       ftpusers     lpc          rc.new       shells
adm          getty        magic        rc0.d        startcons
bcheckrc     gettydefs    motd         rc1.d        swapoff
brc          group        mount        rc2.d        swapon
brc~         inet         mtab         rc3.d        syslog.conf
csh.cshrc    init         mtools       rc4.d        syslog.pid
csh.login    init.d       pac          rc5.d        syslogd.reload
default      initrunlvl   passwd       rmt          termcap
disktab      inittab      printcap     rpc          umount
fdprm        inittab.old  profile      rpcinfo      update
fstab        issue        psdatabase   securetty    utmp
ftpaccess    lilo         rc           services     wtmp
/home/larry#
```

(Para los usuarios de MS-DOS, nótese que los nombres de los ficheros pueden ser mayores de 8 caracteres y pueden contener puntos en cualquier posición. Incluso es posible que un fichero contenga más de un punto en su nombre.)

Vayamos al directorio raíz con "`cd ..`" y desde allí vayamos al directorio `/usr/bin`.

```
/home/larry# cd ..
/home# cd ..
/# cd usr
/usr# cd bin
/usr/bin#
```

También podemos movernos dentro de directorios en múltiples pasos, como en `cd /usr/bin`.

Trate de moverse por varios directorios usando `ls` y `cd`. En algunos casos podrá encontrarse el desagradable mensaje de error "`Permission denied`". Esto simplemente es debido a cuestiones de seguridad del UNIX. Para poder moverse o listar un directorio debe de tener permisos para poder hacerlo. Hablaremos más sobre ello en la Sección 3.9.

3.3.3 Creando directorios nuevos

Es el momento de aprender a crear directorios. Para ello se usa la orden `mkdir`. Pruebe lo siguiente:

```
/home/larry# mkdir foo
/home/larry# ls -F
Mail/
foo/
letters/
papers/
/home/larry# cd foo
/home/larry/foo# ls
/home/larry/foo#
```

¡Enhorabuena! Acaba de crear un directorio nuevo y moverse a él. Como no hay ningún fichero en el directorio nuevo, veamos como copiar ficheros desde un lugar a otro.

3.3.4 Copia de ficheros

La copia de ficheros es efectuada por la orden `cp`:

```
/home/larry/foo# cp /etc/termcap .
/home/larry/foo# cp /etc/shells .
/home/larry/foo# ls –F
shells     termcap
```

```
/home/larry/foo# cp shells bells
/home/larry/foo# ls –F
bells     shells     termcap
/home/larry/foo#
```

La orden `cp` copia los ficheros listados en la línea de comandos al fichero o directorio pasado como último argumento. Nótese como se usa el directorio "`.`" para referirnos al directorio actual.

3.3.5 Moviendo ficheros

La orden `mv` mueve ficheros en lugar de copiarlos. La sintaxis es muy sencilla.

```
/home/larry/foo# mv termcap sells
/home/larry/foo# ls -F
bells     sells     shells
/home/larry/foo#
```

Nótese como `termcap` ya no existe, en su lugar está el fichero `sells`. Esta orden puede usarse para renombrar ficheros, como acabamos de hacer, pero también para mover ficheros a directorios diferentes.

◇ **Nota:** `mv` y `cp` sobreescribirán los ficheros destino (si ya existen) sin consultar. Sea cuidadoso cuando mueva un fichero a otro directorio: puede haber ya un fichero con el mismo nombre que será sobreescrito.

3.3.6 Borrando ficheros y directorios

Para borrar un fichero, use la orden `rm`. ("`rm`" viene de "remove").

```
/home/larry/foo# rm bells sells
/home/larry/foo# ls -F
shells
/home/larry/foo#
```

Nos hemos quedado solo con el fichero "shells", pero no nos quejaremos. Nótese que `rm` por defecto no preguntará antes de borrar un fichero—luego, sea cuidadoso.

Una orden relacionada con `rm` es `rmdir`. Esta orden borra un directorio, pero solo si está vacio. Si el directorio contiene ficheros o subdirectorios, `rmdir` se quejará.

3.3.7 Mirando los ficheros

Las órdenes `more` y `cat` son usadas para ver el contenido de ficheros. `more` muestra el fichero pantalla a pantalla mientras que `cat` lo muestra entero de una vez.

Para ver el contenido del fichero `shells` podemos usar la orden

```
/home/larry/foo# more shells
```

Por si está interesado en el contenido de `shells`, es una lista de intérpretes de comandos válidos disponibles en el sistema. En la mayoría de los sistemas incluye `/bin/sh`, `/bin/bash` y `/bin/csh`. Hablaremos sobre los diferentes intérpretes de comandos más adelante.

Durante la ejecución de `more` pulse `Space` para avanzar a la página siguiente y `b` para volver a la página anterior. Hay otros comandos disponibles, los citados son solo los más básicos. `q` finalizará la ejecución de `more`.

Salga de `more` y pruebe `cat /etc/termcap`. El texto probablemente pasará demasiado rápido como para poder leerlo. El nombre "`cat`" viene de "concatenate", que es para lo que realmente sirve el programa. La orden `cat` puede ser usada para concatenar el contenido de varios ficheros y guardar el resultado en otro fichero. Esto se discutirá más adelante.

3.3.8 Obteniendo ayuda en línea

Prácticamente cada sistema UNIX, incluido Linux, proporciona una utilidad conocida como "páginas de manual". Estas páginas contienen documentación en línea para todas las órdenes del sistema, recursos, ficheros de configuración, etc.

La orden usada para acceder a las páginas de manual es `man`. Por ejemplo, si está interesado en conocer otras opciones de la orden `ls`, puede escribir

```
/home/larry# man ls
```

y le será mostrada la página de manual para `ls`.

Desafortunadamente la mayoría de las páginas de manual han sido escritas por gente que ya conocía lo que la orden o recurso hacía, por esto, las páginas de manual usualmente solo contienen detalles técnicos de la orden sin ningún tipo de tutorial de uso. Pese a esto, estas páginas son una gran fuente de información que permiten refrescar la memoria si olvidamos la sintaxis de un comando. Igualmente, estas páginas le darán mucha información sobre órdenes que no trataremos en este libro.

Le sugiero que pruebe `man` con los comandos que ya hemos tratado y con los que vayamos introduciendo. Notará que alguno de los comandos no tiene página de manual. Esto puede ser debido a diferentes motivos. En primer lugar, las páginas no han sido escritas aún (el Proyecto de Documentación de Linux es también el responsable de las páginas de manual). En segundo lugar, la órden puede ser interna del intérprete de comandos, o un alias (como los tratados en la Sección 3.2.4), en cuyo caso no tendrán una página propia. Un ejemplo es la orden `cd` la cual es interna del intérprete de comandos. El propio intérprete de comandos es quien procesa `cd`—no hay un programa separado.

3.4 Sumario de Ordenes Básicas

Esta sección introduce algunos de las órdenes básicas más útiles de un sistema UNIX, incluidas las ya cubiertas en las secciones anteriores.

Nótese que las opciones usualmente comienzan con "`-`" y en la mayoría de los casos se pueden añadir múltiples opciones de una letra con un único "`-`". Por ejemplo, en lugar de usar `ls -l -F` es posible usar `ls -lF`.

En lugar de listar todas las opciones disponibles para cada uno de los comandos solo hablaremos de aquellas más útiles o importantes. De hecho, la mayoría de las órdenes tienen un gran número de opciones (muchas de las cuales nunca usará). Puede usar `man` para ver las páginas de manual de cada orden, la cual mostrará la lista completa de opciones disponibles.

Nótese también, que la mayoría de las órdenes toman una lista de ficheros o directorios como argumentos, denotados como "⟨*fichero1*⟩ ... ⟨*ficheroN*⟩". Por ejemplo, la orden `cp` toma como argumentos la lista de ficheros a copiar, seguidos del fichero o directorio destino. Cuando se copia más de un fichero, el destino debe de ser un directorio.

`cd` Cambia el directorio de trabajo actual.
Sintaxis: `cd` ⟨*directorio*⟩
⟨*directorio*⟩ es el directorio al que cambiamos. ("`.`" se refiere al directorio actual, "`..`" al directorio padre.)
Ejemplo: `cd ../foo` pone `../foo` como directorio actual.

`ls` Muestra información sobre los ficheros o directorios indicados.
Sintaxis: `ls` ⟨*fichero1*⟩ ⟨*fichero2*⟩ ...⟨*ficheroN*⟩
Donde ⟨*fichero1*⟩ a ⟨*ficheroN*⟩ son los ficheros o directorios a listar.
Opciones: Hay más opciones de las que podría suponer. Las más usadas comúnmente son: `-F` (usada para mostrar información sobre el tipo de fichero), y `-l` (da un listado "largo" incluyendo tamaño, propietario, permisos, etc. Trataremos esto en detalle más adelante.)
Ejemplo: `ls -lF /home/larry` mostrará el contenido del directorio `/home/larry`.

`cp` Copia fichero(s) en otro fichero o directorio.
Sintaxis: `cp` ⟨*fichero1*⟩ ⟨*fichero2*⟩ ...⟨*ficheroN*⟩ ⟨*destino*⟩
Donde ⟨*fichero1*⟩ a ⟨*ficheroN*⟩ son los ficheros a copiar, y ⟨*destino*⟩ es el fichero o directorio destino.
Ejemplo: `cp ../frog joe` copia el fichero `../frog` al fichero o directorio `joe`.

`mv` Mueve fichero(s) a otro fichero o directorio. Es equivalente a una copia seguida del borrado del original. Puede ser usado para renombrar ficheros, como el comando MS-DOS `RENAME`.
Sintaxis: `mv` ⟨*fichero1*⟩ ⟨*fichero2*⟩ ...⟨*ficheroN*⟩ ⟨*destino*⟩
Donde ⟨*fichero1*⟩ a ⟨*ficheroN*⟩ son los ficheros a "mover" y ⟨*destination*⟩ es el fichero o directorio destino.
Ejemplo: `mv ../frog joe` mueve el fichero `../frog` al fichero o directorio `joe`.

`rm` — Borra ficheros. Nótese que cuando los ficheros son borrados en UNIX, son irrecuperables (a diferencia de MS-DOS, donde usualmente se puede recuperar un fichero borrado).
Sintaxis: `rm` ⟨*fichero1*⟩ ⟨*fichero2*⟩ ...⟨*ficheroN*⟩
Donde ⟨*fichero1*⟩ a ⟨*ficheroN*⟩ son los nombres de los ficheros a borrar.
Opciones: `-i` pedirá confirmación antes de borrar un fichero. Ejemplo: `rm -i /home/larry/joe /home/larry/frog` borra los ficheros `joe` y `frog` en `/home/larry`.

`mkdir` — Crea directorios nuevos.
Sintaxis: `mkdir` ⟨*dir1*⟩ ⟨*dir2*⟩ ...⟨*dirN*⟩
Donde ⟨*dir1*⟩ a ⟨*dirN*⟩ son los directorios a crear.
Ejemplo: `mkdir /home/larry/test` crea el directorio `test` colgando de `/home/larry`.

`rmdir` — Esta orden borra directorios vacios. Al usar `rmdir`, el directorio de trabajo actual no debe de estar dentro del directorio a borrar.
Sintaxis: `rmdir` ⟨*dir1*⟩ ⟨*dir2*⟩ ...⟨*dirN*⟩
Donde ⟨*dir1*⟩ a ⟨*dirN*⟩ son los directorios a borrar.
Ejemplo: `rmdir /home/larry/papers` borra el directorio `/home/larry/papers` si está vacio.

`man` — Muestra la página de manual del comando o recurso (cualquier utilidad del sistema que no es un comando, como funciones de librería) dado. Sintaxis: `man` ⟨*command*⟩
Donde ⟨*command*⟩ es el nombre del comando o recurso sobre el que queremos obtener la ayuda.
Ejemplo: `man ls` muestra ayuda sobre la orden `ls`.

`more` — Muesta el contenido de los ficheros indicados, una pantalla cada vez.
Sintaxis: `more` ⟨*fichero1*⟩ ⟨*fichero2*⟩ ...⟨*ficheroN*⟩
Donde ⟨*fichero1*⟩ a ⟨*ficheroN*⟩ son los ficheros a mostrar.
Ejemplo: `more papers/history-final` muestra por el terminal el contenido del fichero `papers/history-final`.

`cat` — Oficialmente usado para concatenar ficheros, `cat` también es usado para mostrar el contenido completo de un fichero de una vez.
Sintaxis: `cat` ⟨*fichero1*⟩ ⟨*fichero2*⟩ ...⟨*ficheroN*⟩
Donde ⟨*fichero1*⟩ a ⟨*ficheroN*⟩ son los ficheros a mostrar.
Ejemplo: `cat letters/from-mdw` muestra por el terminal el contenido del fichero `letters/from-mdw`.

`echo` — Simplemente envía al terminal los argumentos pasados.
Sintaxis: `echo` ⟨*arg1*⟩ ⟨*arg2*⟩ ...⟨*argN*⟩
Donde ⟨*arg1*⟩ a ⟨*argN*⟩ son los argumentos a mostrar.
Ejemplo: `echo ‘‘Hola mundo’’` muestra la cadena “`Hola mundo`”.

`grep` — Muestra todas las líneas de un fichero dado que coinciden con un cierto patrón.

Sintaxis: `grep` ⟨*patrón*⟩ ⟨*fichero1*⟩ ⟨*fichero2*⟩ ...⟨*ficheroN*⟩
Donde ⟨*patrón*⟩ es una expresión regular y ⟨*fichero1*⟩ a ⟨*ficheroN*⟩ son los ficheros donde buscar. Ejemplo: `grep loomer /etc/hosts` mostrará todas las líneas en el fichero `/etc/hosts` que contienen la cadena "`loomer`".

3.5 Explorando el Sistema de Ficheros

El **sistema de ficheros** es la colección de ficheros y la jerarquía de directorios de su sistema. Le prometimos acompañarle por el sistema de ficheros, y ha llegado el momento.

Tiene el nivel y conocimientos para entender de lo que estamos hablando, además de una guía de carreteras. (Figura 3.1 en la página 85).

Primero cambie al directorio raíz (`cd /`) y ejecute `ls -F`. Probablemente verá estos directorios[2]: `bin`, `dev`, `etc`, `home`, `install`, `lib`, `mnt`, `proc`, `root`, `tmp`, `user`, `usr`, y `var`.

Echemos un vistazo a cada uno de estos directorios.

`/bin` — `/bin` es la abreviación de "binaries", o ejecutables. Es donde residen la mayoría de los programas esenciales del sistema. Use la orden "`ls -F /bin`" para listar los ficheros. Podrá ver algunas órdenes que reconocerá, como `cp`, `ls` y `mv`. Estos son los programas para estas órdenes. Cuando usa la orden `cp` está ejecutando el programa `/bin/cp`.

Usando `ls -F` verá que la mayoría (si no todos) los ficheros de `/bin` tienen un asterisco ("`*`") añadido al final de sus nombres. Esto indica que son ficheros ejecutables, como describe la Sección 3.3.2.

`/dev` — El sigiente es `/dev`. Echémosle un vistazo de nuevo con `ls -F`.

Los "ficheros" en `/dev` son conocidos como **controladores de dispositivo (device drivers)**—son usados para acceder a los dispositivos del sistema y recursos, como discos duros, modems, memoria, etc. Por ejemplo, de la misma forma que puede leer datos de un fichero, puede leerla desde la entrada del ratón leyendo `/dev/mouse`.

Los ficheros que comienzan su nombre con `fd` son controladores de disqueteras. `fd0` es la primera disquetera, `fd1` la segunda. Ahora, alguien astuto se dará cuenta de que hay más controladores de dispositivo para disqueteras de los que hemos mencionado. Estos representan tipos específicos de discos. Por ejemplo, `fd1H1440` accederá a discos de 3.5" de alta densidad en la disquetera 1.

Aquí tenemos una lista de algunos de los controladores de dispositivo más usados. Nótese que incluso aunque puede que no tenga alguno de los dispositivos listados, tendrá entradas en `dev` de cualquier forma.

- `/dev/console` hace referencia a la consola del sistema— es decir, al monitor conectado directamente a su sistema.

[2]Puede ver otros o incluso no ver todos. No se preocupe. Cada versión de Linux difiere en algunos aspectos.

- Los dispositivos /dev/ttyS y /dev/cua son usados para acceder a los puertos serie. Por ejemplo, /dev/ttyS0 hace referencia a "COM1" bajo MS-DOS. Los dispositivos /dev/cua son "callout", los cuales son usados en conjunción con un modem.
- Los nombres de dispositivo que comienzan por hd acceden a discos duros. /dev/hda hace referencia a la *totalidad* del primer disco duro, mientras que /dev/hda1 hace referencia a la primera *partición* en /dev/hda.
- Los nombres de dispositivo que comienzan con sd son dispositivos SCSI. Si tiene un disco duro SCSI, en lugar de acceder a él mediante /dev/hda, deberá acceder a /dev/sda. Las cintas SCSI son accedidas vía dispositivos st y los CD-ROM SCSI vía sr.
- Los nombres que comienzan por lp acceden a los puertos paralelo. /dev/lp0 hace referencia a "LPT1" en el mundo MS-DOS.
- /dev/null es usado como "agujero negro"— cualquier dato enviado a este dispositivo desaparece. ¿Para qué puede ser útil esto?. Bien, si desea suprimir la salida por pantalla de una orden, podría enviar la salida a /dev/null. Hablaremos más sobre esto después.
- Los nombres que comienzan por /dev/tty hacen referencia a "consolas virtuales" de su sistema (accesibles mediante las teclas alt-F1, alt-F2, etc). /dev/tty1 hace referencia a la primera VC, /dev/tty2 a la segunda, etc.
- Los nombres de dispositivo que comienzan con /dev/pty son "pseudo-terminales". Estos son usados para proporcionar un "terminal" a sesiones remotas. Por ejemplo, si su máquina está en una red, telnet de entrada usará uno de los dispositivos /dev/pty.

/etc — /etc contiene una serie de ficheros de configuración del sistema. Estos incluyen /etc/passwd (la base de datos de usuarios), /etc/rc (guiones de inicialización del sistema), etc.

/sbin — sbin se usa para almacenar programas esenciales del sistema, que usará el administrador del sistema.

/home — /home contiene los directorios "home" de los usuarios. Por ejemplo, /home/larry es el directorio del usuario "larry". En un sistema recién instalado, no habrá ningún usuario en este directorio.

/lib — /lib contiene las **imágenes de las librerías compartidas**. Estos ficheros contienen código que compartirán muchos programas. En lugar de que cada programa contenga una copia propia de las rutinas compartidas, estas son guardadas en un lugar común, en /lib. Esto hace que los programas ejecutables sean menores y reduce el espacio usado en disco.

/proc — /proc es un "sistema de ficheros virtual". Los ficheros que contiene realmente residen en memoria, no en un disco. Hacen referencia a varios **procesos** que corren en el sistema, y le permiten obtener información acerca de que programas y procesos están corriendo en un momento dado. Entraremos en más detalles en la Sección 3.11.1.

`/tmp` Muchos programas tienen la necesidad de generar cierta información temporal y guardarla en un fichero temporal. El lugar habitual para esos ficheros es en `/tmp`.

`/usr` `/usr` es un directorio muy importante. Contienen una serie de subdirectorios que contienen a su vez algunos de los más importantes y útiles programas y ficheros de configuración usados en el sistema.

Los directorios descritos arriba son esenciales para que el sistema esté operativo, pero la mayoría de las cosas que se encuentran en `/usr` son opcionales para el sistema. De cualquier forma, son estas cosas opcionales las que hacen que el sistema sea útil e interesante. Sin `/usr`, tendría un sistema aburrido, solo con programas como `cp` y `ls`. `usr` contiene la mayoría de los paquetes grandes de programas y sus ficheros de configuración.

`/usr/X386` `/usr/X386` contiene el sistema X Window si usted lo instala. El sistema X Window es un entorno gráfico grande y potente el cual proporciona un gran número de utilidades y programas gráficos, mostrados en "ventanas" en su pantalla. Si está familiarizado con los entornos Microsoft Windows o Macintosh, X Window le será muy familiar. El directorio `/usr/X386` contiene todos los ejecutables de X Window, ficheros de configuración y de soporte. Esto será cubierto con más detalle en la Sección 5.1.

`/usr/bin` `/usr/bin` es el almacén real de programas del sistema UNIX. Contiene la mayoría de los programas que no se encuentran en otras partes como `/bin`.

`/usr/etc` Como `/etc` contiene diferentes ficheros de configuración y programas del sistema, `/usr/etc` contiene incluso más que el anterior. En general, los ficheros que se encuentran en `/usr/etc/` no son esenciales para el sistema, a diferencia de los que se encuentran en `/etc`, que si lo son.

`/usr/include` `/usr/include` contiene los **ficheros de cabecera** para el compilador de C. Estos ficheros (la mayoría de los cuales terminan en `.h`, de "header") declaran estructuras de datos, subrutinas y constantes usados en la escritura de programas en C. Los ficheros que se encuentran en `/usr/include/sys` son generalmente usados en la programación de UNIX a nivel de sistema. Si está familiarizado con el lenguaje de programación C, aquí encontrará los ficheros de cabecera como `stdio.h`, el cual declara funciones como `printf()`.

`/usr/g++-include`

`/usr/g++-include` contiene ficheros de cabecera para el compilador de C++ (muy parecido a `/usr/include`).

`/usr/lib` `/usr/lib` contiene las librerías equivalentes "stub" y "static" a los ficheros encontrados en `/lib`. Al compilar un programa, este es "enlazado" con las librerías que se encuentran en `/usr/lib`, las cuales dirigen al programa a buscar en `/lib` cuando necesita el código de la librería. Además, varios programas guardan ficheros de configuración en /usr/lib.

/usr/local — /usr/local es muy parecido a /usr—contiene programas y ficheros no esenciales para el sistema, pero que hacen el sistema más divertido y excitante. En general, los programas que se encuentran en /usr/local son específicos de su sistema—esto es, el directorio /usr/local difiere bastante entre sistemas UNIX.

Aquí encontrará programas grandes como TEX (sistema de formateo de documentos) y Emacs (gran y potente editor), si los instala.

/usr/man — Este directorio contiene las páginas de manual. Hay dos subdirectorios para cada página "sección" de las páginas (use la orden man man para más detalles). Por ejemplo, /usr/man/man1 contiene los fuentes (es decir, los originales por formatear) de las páginas de manual de la sección 1, y /usr/man/cat1 las páginas ya formateadas de la sección 1.

/usr/src — /usr/src contiene el código fuente (programas por compilar) de varios programas de su sistema. El más importante es /usr/src/linux, el cual contiene el código fuente del Núcleo de Linux.

/var — /var contiene directorios que a menudo cambian su tamaño o tienden a crecer. Muchos de estos directorios solian residir en /usr, pero desde que estamos tratando de dejarlo relativamente inalterable, los directorios que cambian a menudo han sido llevados a /var. Algunos de estos directorios son:

/var/adm — /var/adm contiene varios ficheros de interés para el administrador del sistema, especificamente históricos del sistema, los cuales recogen errores o problemas con el sistema. Otros ficheros guardan las sesiones de presentación en el sistema, así como los intentos fallidos. Esto será cubierto en el Capítulo 4.

/var/spool — /var/spool contiene ficheros que van a ser pasados a otro programa. Por ejemplo, si su máquina está conectada a una red, el correo de llegada será almacenado en /var/spool/mail hasta que lo lea o lo borre. Artículos nuevos de las "news" tanto salientes como entrantes pueden encontrarse en /var/spool/news, etc.

3.6 Tipos de intérpretes de comandos

Como hemos mencionado anteriormente en numerosas ocasiones, UNIX es un sistema operativo multitarea y multiusuario. La multitarea es *muy* útil, y una vez la haya probado, la usará continuamente. En poco tiempo podrá ejecutar programas "de fondo", conmutar entre múltiples tareas y "entubar" programas unos entre otros para conseguir resultados complejos con un único comando.

Muchas de las características que trataremos en esta sección son proporcionadas por el intérprete de comandos. Hay que tener cuidado en no confundir UNIX (el sistema operativo) con el intérprete de comandos– este último, es un interface con el sistema que hay debajo. El intérprete de comandos proporciona la funcionalidad sobre el UNIX.

El intérprete de comandos no es solo un intérprete interactivo de los comandos que tecleamos, es también un potente lenguaje de programación, el cual permite escribir **guiones**, que permiten juntar varias órdenes en un fichero. Los usuarios de MS-DOS reconoceran esto como los ficheros "batch". El uso de los guiones del intérprete de comandos es una herramienta muy potente que le permitirá automatizar e incrementar el uso de UNIX. Ver la sección 3.13.1 para más información.

Hay varios tipos de intérpretes de comandos en el mundo UNIX. Los dos más importantes son el "Bourne shell" y el "C shell". El intérprete de comandos Bourne, usa una sintaxis de comandos como la usada en los primeros sistemas UNIX, como el System III. El nombre del intérprete Bourne en la mayoría de los UNIX es `/bin/sh` (donde `sh` viene de "shell", intérprete de comandos en inglés). El intérprete C usa una sintaxis diferente, a veces parecida a la del lenguaje de programación C, y en la mayoría de los sistemas UNIX se encuentra como `/bin/csh`.

Bajo Linux hay algunas diferencias en los intérpretes de comandos disponibles. Dos de los más usados son el "Bourne Again Shell" o "Bash" (`/bin/bash`) y Tcsh (`/bin/tcsh`). Bash es un equivalente al Bourne con muchas características avanzadas de la C shell. Como Bash es un superconjunto de la sintaxis del Bourne, cualquier guión escrito para el intérprete de comandos Bourne standard funcionará en Bash. Para los que prefieren el uso del intérprete de comandos C, Linux tiene el Tcsh, que es una versión extendida del C original.

El tipo de intérprete de comandos que decida usar es puramente una cuestión de gustos. Algunas personas prefieren la sintaxis del Bourne con las características avanzadas que proporciona Bash, y otros prefieren el más estructurado intérprete de comandos C. En lo que respecta a los comandos usuales como `cp`, `ls`..etc, es indiferente el tipo de intérprete de comandos usado, la sintaxis es la misma. Solo, cuando se escriben guiones para el intérprete de comandos, o se usan características avanzadas aparecen las diferencias entre los diferentes intérpretes de comandos.

Como estamos discutiendo sobre las diferencias entre los intérpretes de comandos Bourne y C, abajo veremos esas diferencias. Para los propósitos de este manual, la mayoría de las diferencias son mínimas. (Si eres realmente curioso a este respecto, lee las páginas de manual para `bash` y `tcsh`).

3.7 Carácteres comodín

Una característica importante de la mayoría de los intérpretes de comandos en Unix es la capacidad para referirse a más de un fichero usando carácteres especiales. Estos llamados **comodines** le permiten referirse a, por ejemplo, todos los ficheros que contienen el carácter "`n`".

El comodín "`*`" hace referencia cualquier carácter o cadena de carácteres en el fichero. Por ejemplo, cuando usa el carácter "`*`" en el nombre de un fichero, el intérprete de comandos lo sustituye por todas las combinaciones posibles provenientes de los ficheros en el directorio al cual nos estamos refiriendo.

Veamos un ejemplo rápido. Supongamos que Larry tiene los ficheros `frog`, `joe` y `stuff` en el directorio actual.

```
/home/larry# ls
```

```
frog     joe     stuff
/home/larry#
```

Para aceder a todos los ficheros con la letra "o" en su nombre, hemos de usar la orden

```
/home/larry# ls *o*
frog     joe
/home/larry#
```

Como puede ver, el comodín "*" ha sido sustituido con todas las combinaciones posibles que coincidian de entre los ficheros del directorio actual.

El uso de "*" solo, simplemente se refiere a todos los ficheros, puesto que todos los carácteres coinciden con el comodín.

```
/home/larry# ls *
frog     joe     stuff
/home/larry#
```

Veamos unos pocos ejemplos más.

```
/home/larry# ls f*
frog
/home/larry# ls *ff
stuff
/home/larry# ls *f*
frog     stuff
/home/larry# ls s*f
stuff
/home/larry#
```

El proceso de la sustitución de "*" en nombres de ficheros es llamado **expansión de comodines** y es efectuado por el intérprete de comandos. Esto es importante: las órdenes individuales, como `ls`, *nunca* ven el "*" en su lista de parámetros. Es el intérprete quien expande los comodines para incluir todos los nombres de ficheros que se adaptan. Luego la orden

```
/home/larry# ls *o*
```

es expandida para obtener

```
/home/larry# ls frog joe
```

Una nota importante acerca del carácter comodín "*". El uso de este comodín *NO* cuadrará con nombres de ficheros que comiencen con un punto ("`.`"). Estos ficheros son tratados como "ocultos"— aunque no están realmente ocultos, simplemente no son mostrados en un listado normal de `ls` y no son afectados por el uso del comodín "*".

He aquí un ejemplo. Ya hemos mencionado que cada directorio tiene dos entradas especiales: "." que hace referencia al directorio actual y ".." que se refiere al directorio padre. De cualquier forma, cuando use `ls` esas dos entradas no se mostrarán.

```
/home/larry# ls
frog     joe     stuff
/home/larry#
```

Si usa el parámetro `-a` con `ls` podrá ver nombres de ficheros que comienzan por ".". Observe:

```
/home/larry# ls -a
.     ..     .bash_profile     .bashrc     frog     joe     stuff
/home/larry#
```

Ahora podemos ver las dos entradas especiales, "." y "..", así como otros dos ficheros "ocultos"— `.bash_profile` y `.bashrc`. Estos dos ficheros son usados en el arranque por `bash` cuando larry se presenta al sistema. Más información sobre esto en la Sección 3.13.3.

Note que cuando usamos el comodín "`*`", no se muestra ninguno de los nombres de fichero que comienzan por ".".

```
/home/larry# ls *
frog     joe     stuff
/home/larry#
```

Esto es una característica de seguridad: si "`*`" coincidiera con ficheros que comienzan por "." actuaría sobre "." y "..". Esto puede ser peligroso con ciertas órdenes.

Otro carácter comodín es "`?`". Este carácter comodín solo expande un único carácter. Luego "`ls ?`" mostrará todos los nombres de ficheros con un carácter de longitud, y "`ls termca?`" mostrará "`termcap`" pero *no* "`termcap.backup`". Aquí tenemos otro ejemplo:

```
/home/larry# ls j?e
joe
/home/larry# ls f??g
frog
/home/larry# ls ????f
stuff
/home/larry#
```

Como puede ver, los carácteres comodín le permiten referirse a más de un fichero a la vez. En el resumen de órdenes en la Sección 3.4 dijimos que `cp` y `mv` pueden copiar o mover múltiples ficheros de una vez. Por ejemplo,

```
/home/larry# cp /etc/s* /home/larry
```

copiará todos los ficheros de `/etc` que comiencen por "`s`" al directorio `/home/larry`. Por lo tanto, el formato de la orden `cp` es realmente

`cp` ⟨*fichero1*⟩ ⟨*fichero2*⟩ ⟨*fichero3*⟩ ...⟨*ficheroN*⟩ ⟨*destino*⟩

donde ⟨*fichero1*⟩ a ⟨*ficheroN*⟩ es la lista de los ficheros a copiar, y ⟨*destino*⟩ es el fichero o directorio destino donde copiarlos. `mv` tiene idéntica sintaxis.

Notese que si está copiando o moviendo más de un fichero, ⟨*destino*⟩ debe ser un directorio. Solo puede copiar o mover un *único* fichero a otro fichero.

3.8 Fontanería UNIX

3.8.1 Entrada y salida estandard

Muchos comandos UNIX toman su entrada de algo conocido como **entrada estandard** y envían su salida a la **salida estándar** (a menudo abreviado como "stdin" y "stdout"). El intérprete de comandos configura el sistema de forma que la entrada estándar es el teclado y la salida la pantalla.

Veamos un ejemplo con el comando `cat`. Normalmente `cat` lee datos de los ficheros cuyos nombres se pasan como argumentos en la línea de comandos y envía estos datos directamente a la salida estándar. Luego, usando el comando

```
/home/larry/papers# cat history-final masters-thesis
```

mostrará por pantalla el contenido del fichero `history-final` seguido por `masters-thesis`.

Si no se le pasan nombres de ficheros a `cat` como parámetros, leerá datos de stdin y los enviará a stdout. Veamos un ejemplo.

```
/home/larry/papers# cat
Hello there.
Hello there.
Bye.
Bye.
ctrl-D
/home/larry/papers#
```

Como se puede ver, cada línea que el usuario teclea (impresa en itálica) es inmediatamente reenviada al monitor por `cat`. Cuando se está leyendo de la entrada estándar, los comandos reconocen el fin de la entrada de datos cuando reciben el carácter EOT (end-of-text, fin de texto). Normalmente es generado con la combinación `ctrl-D`.

Veamos otro ejemplo. El comando `sort` toma como entrada líneas de texto (de nuevo leerá desde stdin si no se le proporcionan nombres de ficheros en la línea de comandos), y devuelve la salida ordenada a stdout. Pruebe lo siguiente:

```
/home/larry/papers# sort
bananas
carrots
apples
ctrl-D
apples
bananas
carrots
/home/larry/papers#
```

Podemos ordenar alfabéticamente la lista de la compra... ¿no es útil UNIX?

3.8.2 Redireccionando la entrada y salida

Ahora, supongamos que queremos que la salida de `sort` vaya a un fichero para poder salvar la lista ordenada de salida. El intérprete de comandos nos permite **redireccionar** la salida estándar a un fichero usando el símbolo "`>`". Veamos como funciona.

```
/home/larry/papers# sort > shopping-list
bananas
carrots
apples
ctrl-D
/home/larry/papers#
```

Como puede ver, el resultado de `sort` no se muestra por pantalla, en su lugar es salvado en el fichero `shopping-list`. Echemos un vistazo al fichero.

```
/home/larry/papers# cat shopping-list
apples
bananas
carrots
/home/larry/papers#
```

Ya podemos ordenar la lista de la compra y además guardarla.

Supongamos ahora que teníamos guardada nuestra lista de compra desordenada original en el fichero `items`. Una forma de ordenar la información y salvarla en un fichero podría ser darle a `sort` el nombre del fichero a leer en lugar de la entrada estándar y redireccionar la salida estándar como hicimos arriba.

```
/home/larry/papers# sort items > shopping-list
/home/larry/papers# cat shopping-list
apples
bananas
```

```
carrots
/home/larry/papers#
```

Hay otra forma de hacer esto. No solo puede ser redireccionada la salida estándar, también puede ser redireccionada la *entrada* estándar usando el símbolo “<”.

```
/home/larry/papers# sort < items
apples
bananas
carrots
/home/larry/papers#
```

Técnicamente, `sort < items` es equivalente a `sort items`, pero nos permite demostrar que `sort < items` se comporta como si los datos del fichero fueran tecleados por la entrada estándar. El intérprete de comandos es quien maneja las redirecciones. `sort` no recibe el nombre del fichero (`items`) a leer, desde el punto de vista de `sort`, está leyendo datos de la entrada estándar como si fueran tecleados desde el teclado.

Esto introduce el concepto de **filtro**. Un filtro es un programa que lee datos de la entrada estándar, los procesa de alguna forma, y devuelve los datos procesados por la salida estándar. Usando la redirección la entrada estándar y/o salida estándar pueden ser referenciadas desde ficheros. `sort` es un filtro simple: ordena los datos de entrada y envía el resultado a la salida estándar. `cat` es incluso más simple, no hace nada con los datos de entrada, simplemente envía a la salida cualquier cosa que le llega.

3.8.3 Uso de tuberías (pipes)

Ya hemos visto como usar `sort` como un filtro. Pero estos ejemplos suponen que tenemos los datos en un fichero en alguna parte o vamos a introducir los datos manualmente por la entrada estándar. ¿Qué pasa si los datos que queremos ordenar provienen de la salida de otro comando, como `ls`?. Por ejemplo, usando la opción `-r` con `sort` ordenaremos los datos en orden inverso. Si queremos listar los ficheros en el directorio actual en orden inverso, una forma podría ser.

```
/home/larry/papers# ls
english-list
history-final
masters-thesis
notes
/home/larry/papers# ls > file-list
/home/larry/papers# sort -r file-list
notes
masters-thesis
history-final
english-list
/home/larry/papers#
```

Aquí, salvamos la salida de `ls` en un fichero, y entonces ejecutamos `sort -r` sobre ese fichero. Pero esta forma necesita crear un fichero temporal en el que salvar los datos generados por `ls`.

La solución es usar las **pipes**[3]. El uso de pipes es otra característica del intérprete de comandos, que nos permite conectar una cadena de comandos en un "pipe", donde la stdout del primero es enviada directamente a la stdin del segundo y así sucesivamente. Queremos conectar la salida de `ls` con la entrada de `sort`. Para crear un pipe se usa el símbolo "`|`":

```
/home/larry/papers# ls | sort -r
notes
masters-thesis
history-final
english-list
/home/larry/papers#
```

Esta forma es más corta y obviamente más fácil de escribir.

Otro ejemplo útil— usando el comando

```
/home/larry/papers# ls /usr/bin
```

mostrará una lista larga de los ficheros, la mayoría de los cuales pasará rápidamente ante nuestros ojos sin que podamos leerla. En lugar de esto, usemos `more` para mostrar la lista de ficheros en `/usr/bin`.

```
/home/larry/papers# ls /usr/bin | more
```

Ahora podemos ir avanzando página a página comodamente.

¡Pero la diversión no termina aquí!. Podemos "entubar" más de dos comandos a la vez. El comando `head` es un filtro que muestra la primeras líneas del canal de entrada (aquí la entrada desde una pipe). Si queremos ver el último fichero del directorio actual en orden alfabético, usaremos:

```
/home/larry/papers# ls | sort -r | head -1
notes
/home/larry/papers#
```

Donde `head -1` simplemente muestra la primera línea de la entrada que recibe en este caso, el flujo de datos ordenados inversamente provenientes de `ls`).

3.8.4 Redirección no destructiva

El uso de ">" para redireccionar la salida a un fichero es destructivo: en otras palabras, el comando

```
/home/larry/papers# ls > file-list
```

[3] N. del T.: tuberías

sobreescribe el contenido del fichero `file-list`. Si en su lugar, usamos el símbolo "`>>`", la salida será añadida al final del fichero nombrado, en lugar de ser sobreescrito.

`/home/larry/papers#` *ls >> file-list*

añadirá la salida de `ls` al final de `file-list`.

Es conveniente tener en cuenta que la redirección y el uso de pipes son características proporcionadas por el intérprete de comandos— este, proporciona estos servicios mediante el uso de la sintaxis "`>`", "`>>`" y "`|`".

3.9 Permisos de Ficheros

3.9.1 Conceptos de permisos de ficheros

Al ser UNIX un sistema multiusuario, para proteger ficheros de usuarios particulares de la manipulación por parte de otros, UNIX proporciona un mecanismo conocido como **permisos de ficheros**. Este mecanismo permite que ficheros y directorios "pertenezcan" a un usuario en particular. Por ejemplo, como Larry creó ficheros en su directorio "home", Larry es el propietario de esos ficheros y tiene acceso a ellos.

UNIX también permite que los ficheros sean compartidos entre usuarios y grupos de usuarios. Si Larry lo desea, podría restringir el acceso a sus ficheros de forma que ningún otro usuario tenga acceso. De cualquier modo, en la mayoría de los sistemas por defecto se permite que otros usuarios puedan leer tus ficheros pero no modificarlos o borrarlos.

Como hemos explicado arriba, cada fichero pertenece a un usuario en particular. Por otra parte, los ficheros también pertenecen a un **grupo** en particular, que es un conjunto de usuarios definido por el sistema. Cada usuario pertenece al menos a un grupo cuando es creado. El administrador del sistema puede hacer que un usuario tenga acceso a más de un grupo.

Los grupos usualmente son definidos por el tipo de usuarios que acceden a la máquina. Por ejemplo, en un sistema UNIX de una universidad, los usuarios pueden ser divididos en los grupos `estudiantes`, `dirección`, `profesores` e `invitados`. Hay también unos pocos grupos definidos por el sistema (como `bin` y `admin`) los cuales son usados por el propio sistema 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. Estos permisos pueden ser fijados para tres clases de usuarios: el propietario del fichero, el grupo al que pertenece el fichero y para todos los usuarios independientemente del grupo.

El permiso de lectura permite a un usuario leer el contenido del fichero o en el caso de un directorio, listar el contenido del mismo (usando `ls`). El permiso 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 dicho directorio. Por último, el permiso de ejecución permite a un usuario ejecutar el fichero si es un programa o guión del intérprete de comandos. Para directorios, el permiso de ejecución permite al usuario cambiar al directorio en cuestión con `cd`.

3.9.2 Interpretando los permisos de ficheros

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, el cual incluye los permisos de ficheros.

```
/home/larry/foo# ls -l stuff

-rw-r--r--   1 larry    users         505 Mar 13 19:05 stuff

/home/larry/foo#
```

El primer campo impreso en el listado representa los permisos de ficheros. El tercer campo es el propietario del fichero (`larry`), y el cuarto es el grupo al cual pertenece el fichero (`users`). Obviamente, el último campo es el nombre del fichero (`stuff`), y los demás campos los trataremos más adelante.

Este fichero pertenece a `larry` y al grupo `users`. Echemos un vistazo a los permisos. La cadena `-rw-r--r--` nos informa, por orden, de los permisos para el propietario, el grupo del fichero y cualquier otro usuario.

El primer carácter de la cadena de permisos ("`-`") representa el tipo de fichero. El "`-`" significa que es un fichero regular. Las siguientes tres letras ("`rw-`") representan los permisos para el propietario del fichero, `larry`. El "`r`" para "lectura" y "`w`" para escritura. Luego `Larry` tiene permisos de lectura y escritura para el fichero `stuff`.

Como ya mencionamos, aparte de los permisos de lectura y escritura está el permiso de "ejecución", representado por una "`x`". Como hay un "`-`" en lugar del "`x`", significa que Larry no tiene permiso para ejecutar ese fichero. Esto es correcto, puesto que `stuff` no es un programa de ningún tipo. Por supuesto, como el fichero es de Larry, puede darse a si mismo permiso de ejecución si lo desea. Esto será cubierto en breve.

Los siguientes tres carácteres, `r--` representan los permisos para los miembros del grupo. El grupo al que pertenece el fichero es `users`. Como solo aparece un "`r`" cualquier usuario que pertenezca al grupo `users` puede leer este fichero.

Las últimos tres carácteres, también `r--`, representan los permisos para cualquier otro usuario del sistema (diferentes del propietario o de los pertenecientes al grupo `users`). De nuevo, como solo está presente el "`r`", los demás usuarios pueden leer el fichero, pero no escribir en él o ejecutarlo.

Aquí tenemos otros ejemplos de permisos de grupo.

`-rwxr-xr-x` El propietario del fichero puede leer, escribir y ejecutar el fichero. Los usuarios pertenecientes al grupo del fichero, y todos los demás usuarios pueden leer y ejecutar el fichero.

`-rw-------` El propietario del fichero puede leer y escribir. Nadie más puede acceder al fichero.

`-rwxrwxrwx` Todos los usuarios pueden leer, escribir y ejecutar el fichero.

3.9.3 Dependencias

Es importante darse cuenta de 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 tengan permiso de lectura y ejecución para el directorio en el cual se encuentra el fichero. Si Larry quiere restringir el acceso a todos sus ficheros, podría simplemente poner los permisos de su directorio "home" `/home/larry` a `-rwx------`. De esta forma ningún usuario podrá acceder a su directorio ni a ninguno de sus ficheros o subdirectorios. Larry no necesita preocuparse de los permisos individuales de cada uno de sus ficheros.

En otras palabras, para acceder a un fichero, debes de 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 en particular.

Habitualmente, los usuarios de un sistema UNIX 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 modificarlos de ninguna forma. Los directorios, usualmente tienen los permisos `-rwxr-xr-x`, lo que permite que los demás usuarios puedan moverse y ver los directorios, pero sin poder crear o borrar nuevos ficheros en ellos.

Muchos usuarios pueden querer limitar el acceso de otros usuarios a sus ficheros. Poniendo los permisos de un fichero a `-rw-------` no se permitirá a ningún otro usuario acceder al fichero. Igualmente, poniendo los permisos del directorio a `-rwx------` no se permitirá a los demás usuarios acceder al directorio en cuestión.

3.9.4 Cambiando permisos

El comando `chmod` se usa para establecer los permisos de un fichero. Solo el propietario puede cambiar los permisos del fichero. La sintaxis de `chmod` es:

`chmod {a,u,g,o}{+,-}{r,w,x}` ⟨*filenames*⟩

Brevemente, indicamos a que usuarios afecta **a**ll, **u**ser, **g**roup o **o**ther. Entonces se especifica si se están añadiendo permisos (`+`) o quitándolos (`-`). Finalmente se especifica que tipo de permiso **r**ead, **w**rite o e**x**ecute. Algunos ejemplos:

`chmod a+r stuff`
: Da a todos los usuarios acceso al fichero.

`chmod +r stuff`
: Como arriba— si no se indica `a`, `u`, `g` o `o` por defecto se toma `a`.

`chmod og-x stuff`
: Quita permisos de ejecución a todos los usuarios excepto al propietario.

`chmod u+rwx stuff`
: Permite al propietario leer, escribir y ejecutar el fichero.

`chmod o-rwx stuff`
Quita permisos de lectura, escritura y ejecución a todos los usuarios menos al propietario y a los usuarios del grupo del fichero.

3.10 Manejando enlaces de ficheros

Los enlaces le permiten dar a un único fichero múltiples nombres. Los ficheros son identificados por el sistema por su **número de inodo**, el cual es el único identificador del fichero para el sistema de ficheros [4]. Un directorio es una lista de números de inodo con sus correspondientes nombres de fichero. Cada nombre de fichero en un directorio es un **enlace** a un inodo particular.

3.10.1 Enlaces duros (Hard links)

La orden `ln` es usada para crear múltiples enlaces para un fichero. Por ejemplo, supongamos que tiene un fichero `foo` en un directorio. Usando `ls -i`, veremos el número de inodo para el fichero.

```
# ls -i foo
22192 foo
#
```

Aqui, el fichero `foo` tiene el número de inodo 22192 en el sistema de ficheros. Podemos crear otro enlace a `foo`, llamado `bar`:

```
# ln foo bar
```

Con `ls -i` veremos que los dos ficheros tienen el mismo inodo.

```
# ls -i foo bar
22192 bar   22192 foo
#
```

Ahora, accediendo a `foo` o a `bar` accederemos al mismo fichero. Si hace cambios en `foo`, estos cambios también serán efectuados en `bar`. Para todos los efectos, `foo` y `bar` son el mismo fichero.

Estos enlaces son conocidos como *enlaces duros (hard links)* porque directamente crean el enlace al inodo. Notese que solo podemos crear enlaces duros entre ficheros del mismo sistema de ficheros; enlaces simbólicos (ver más adelante) no tienen esta restricción.

Cuando borra un fichero con `rm`, está solamente borrando un enlace a un fichero. Si usa el comando

```
# rm foo
```

[4] La orden `ls -i` mostrará los números de inodo.

solo el enlace de nombre `foo` es borrado; `bar` todavía existirá. Un fichero es solo definitivamente borrado del sistema cuando no quedan enlaces a él. Usualmente, los ficheros tienen un único enlace, por lo que el uso de `rm` los borra. Pero si el fichero tiene múltiples enlaces, el uso de `rm` solo borrará un único enlace; para borrar el fichero, deberá borrar todos los enlaces del fichero.

La orden `ls -l` muestra el número de enlaces a un fichero (entre otra información).

```
# ls -l foo bar
-rw-r--r--   2 root     root           12 Aug  5 16:51 bar
-rw-r--r--   2 root     root           12 Aug  5 16:50 foo
#
```

La segunda columna en el listado, "2", especifica el número de enlaces al fichero.

Así resulta que un directorio no es más que un fichero que contiene información sobre la translación enlace a inodo. También, cada directorio tiene al menos dos enlaces duros en él: "`.`" (un enlace apuntando a si mismo) y "`..`" (un enlace apuntando al directorio padre). En el directorio raíz (`/`), el enlace "`..`" simplemente apunta a `/`.

3.10.2 Enlaces simbólicos

Los enlaces simbólicos son otro tipo de enlace, que es diferente al enlace duro. Un enlace simbólico permite dar a un fichero el nombre de otro, pero no enlaza el fichero con un inodo.

La orden `ln -s` crea un enlace simbólico a un fichero. Por ejemplo, si usamos la orden

```
# ln -s foo bar
```

crearemos un enlace simbólico `bar` apuntando al fichero `foo`. Si usamos `ls -i`, veremos que los dos ficheros tienen inodos diferentes, en efecto.

```
# ls -i foo bar
22195 bar   22192 foo
#
```

De cualquier modo, usando `ls -l` vemos que el fichero `bar` es un enlace simbólico apuntando a `foo`.

```
# ls -l foo bar
lrwxrwxrwx   1 root     root            3 Aug  5 16:51 bar -> foo
-rw-r--r--   1 root     root           12 Aug  5 16:50 foo
#
```

Los bits de permisos en un enlace simbólico no se usan (siempre aparecen como `rwxrwxrwx`). En su lugar, los permisos del enlace simbólico son determinados por los permisos del fichero "apuntado" por el enlace (en nuestro ejemplo, el fichero `foo`).

Funcionalmente, los enlaces duros y simbólicos son similares, pero hay algunas diferencias. Por una parte, puede crear un enlace simbólico a un fichero que no existe; lo mismo no es cierto para enlaces duros. Los enlaces simbólicos son procesados por el núcleo de forma diferente a los duros, lo cual es solo una diferencia técnica, pero a veces importante. Los enlaces simbólicos son de ayuda puesto que identifican al fichero al que apuntan; con enlaces duros no hay forma fácil de saber que fichero está enlazado al mismo inodo.

Los enlaces se usan en muchas partes del sistema Linux. Los enlaces simbólicos son especialmente importantes para las imágenes de las librerías compartidas en `/lib`. Ver la Sección 4.7.2 para más información.

3.11 Control de Tareas

3.11.1 Tareas y procesos

Control de Tareas es una utilidad incluida en muchos shells (incluidas Bash y Tcsh), que permite el control de multitud de comandos o **tareas** al momento. Antes de seguir, deberemos hablar un poco sobre los **procesos**.

Cada vez que usted ejecuta un programa, usted lanza lo que se conoce como *proceso*, que es simplemente el nombre que se le da a un programa cuando se esta ejecutando. El comando `ps` visualiza la lista de procesos que se están ejecutando actualmente, por ejemplo:

```
/home/larry# ps

  PID TT STAT  TIME COMMAND
   24  3 S     0:03 (bash)
  161  3 R     0:00 ps

/home/larry#
```

La columna `PID` representa el **identificador de proceso**. La última columna `COMMAND`, es el nombre del proceso que se está ejecutando. Ahora solo estamos viendo los procesos que está ejecutando Larry[5]. Vemos que hay dos procesos, `bash` (Que es el shell o intérprete de comandos que usa Larry), y el propio comando `ps`. Como puede observar, la `bash` se ejecuta concurrentemente con el comando `ps`. La `bash` ejecutó `ps` cuando Larry tecleó el comando. Cuando `ps` termina de ejecutarse (después de mostrar la tabla de procesos), el control retorna al proceso `bash`, que muestra el prompt, indicando que está listo para recibir otro comando.

Un proceso que esta corriendo se denomina *tarea* para el shell. Los términos *proceso* y *tarea*, son intercambiables. Sin embargo, se suele denominar "tarea" a un proceso, cuando es usado en conjunción con **control de tareas**, que es un rasgo del shell que permite cambiar entre distintas tareas.

[5]Hay muchos más procesos aparte de estos corriendo en el sistema, para verlos todos, teclearemos el comando "`ps -aux`".

En muchos casos, los usuarios solo ejecutan un trabajo cada vez, que es el último comando que ellos teclearon desde el shell. Sin embargo, usando el control de tareas, usted podrá ejecutar diferentes tareas al mismo tiempo, cambiando entre cada uno de ellos conforme lo necesite. ¿Cuán beneficioso puede llegar a ser esto?. Supongamos que está usted con su procesador de textos, y de repente necesita parar y realizar otra tarea, con el control de tareas, usted podrá suspender temporalmente el editor, y volver al shell para realizar cualquier otra tarea, y luego regresar al editor como si no lo hubiese dejado nunca. Lo siguiente solo es un ejemplo, hay montones de usos prácticos del control de tareas.

3.11.2 Primer plano y Segundo plano

Un proceso puede estar en **Primer plano** o en **Segundo plano**. Solo puede haber un proceso en primer plano al mismo tiempo, el proceso que está en primer plano, es el que interactúa con usted—recibe entradas de teclado, y envía las salidas al monitor. (Salvo, por supuesto, que haya redirigido la entrada o la salida, como se describe en la Sección 3.8). El proceso en segundo plano, no recibe ninguna señal desde el teclado—por lo general, se ejecutan en silencio sin necesidad de interacción.

Algunos programas necesitan mucho tiempo para terminar, y no hacen nada interesante mientras tanto. Compilar programas es una de estas tareas, así como comprimir un fichero grande. No tiene sentido que se siente y se aburra mientras estos procesos terminan. En estos casos es mejor lanzarlos en segundo plano, para dejar el ordenador en condiciones de ejecutar otro programa.

Los procesos pueden ser **suspendidos**. Un proceso suspendido es aquel que no se está ejecutando actualmente, sino que está temporalmente parado. Después de suspender una tarea, puede indicar a la misma que continúe, en primer plano o en segundo, según necesite. Retomar una tarea suspendida no cambia en nada el estado de la misma—la tarea continuará ejecutandose justo donde se dejó.

Tenga en cuenta que suspender un trabajo no es lo mismo que *interrumpirlo*. Cuando usted interrumpe un proceso (generalmente con la pulsación de `ctrl-C`[6]), el proceso muere, y deja de estar en memoria y utilizar recursos del ordenador. Una vez eliminado, el proceso no puede continuar ejecutandose, y deberá ser lanzado otra vez para volver a realizar sus tareas. También se puede dar el caso de que algunos programas capturan la interrupción, de modo que pulsando `ctrl-C` no se para inmediatamente. Esto se hace para permitir al programa realizar operaciones necesarias de limpieza antes de terminar[7]. De hecho, algunos programas simplemente no se dejan matar por ninguna interrupción.

3.11.3 Envío a segundo plano y eliminación de procesos

Empecemos con un ejemplo sencillo. El comando `yes` es un comando aparentemente inútil que envía una serie interminable de `y`-es a la salida estándar. (Realmente es muy útil. Si se utiliza una tubería

[6] La tecla de interrupción puede definirse usando el comando `stty`. Por defecto, en la mayoría de sistemas es `ctrl-C`, pero no se puede garantizar que sea la misma en su sistema.

[7] Tiempo necesario para guardar algunos registros, etc.

(o "pipe") para unir la salida de `yes` con otro comando que haga preguntas del tipo si/no, la serie de `y`-es confirmará todas las preguntas.)

Pruebe con esto.

```
/home/larry# yes
y
y
y
y
y
```

La serie de `y`-es continuará hasta el infinito, a no ser que usted la elimine, pulsando la tecla de interrupción, generalmente `ctrl-C`. También puede deshacerse de esta serie de `y`-es redigiriendo la salida estándar de `yes` hacia `/dev/null`, que como recordará es una especie de "agujero negro" o papelera para los datos. Todo lo que usted envíe allí, desaparecerá.

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

Ahora va mucho mejor, el terminal no se ensucia, pero el prompt de la shell no retorna. Esto es porque `yes` sigue ejecutandose y enviando esos inútiles `y`-es a `/dev/null`. Para recuperarlo, pulse la tecla de interrupción.

Supongamos ahora que queremos dejar que el comando `yes` siga ejecutandose, y volver al mismo tiempo a la shell para trabajar en otras cosas. Para ello nos enviaremos a `yes` a segundo plano, lo que nos permitirá ejecutarlo, pero sin necesidad de interacción.

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

```
/home/larry# yes > /dev/null &
[1] 164
/home/larry#
```

Como podrá ver, ha regresado a la shell. ¿Pero qué es eso de "`[1] 164`"?, ¿se está ejecutando realmente el comando `yes`?

"`[1]`" representa el **número de tarea** del proceso `yes`. La shell asigna un número a cada tarea que se esté ejecutando. Como `yes` es el único comando que se está ejecutando, se le asigna el número de tarea `1`. El número "`164`" es el número de identificación del proceso, o PID, que es el número que el sistema le asigna al proceso. Ambos números pueden usarse para referirse a la tarea como veremos después.

Ahora usted tiene el proceso `yes` corriendo en segundo plano, y enviando constantemente la señal `y` hacia el dispositivo `/dev/null`. Para chequear el estado del proceso, utilice el comando interno de la shell `jobs`:

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

¡Ahí está!. También puede usar el comando `ps`, como mostramos antes, para comprobar el estado de la tarea.

Para eliminar una tarea, utilice el comando `kill`. Este comando toma como argumento un número de tarea o un número de ID de un proceso. Esta era la tarea 1, así que usando el comando

```
/home/larry# kill %1
```

matará la tarea. Cuando se identifica la tarea con el número de tarea, se debe preceder el número con el carácter de porcentaje ("`%`").

Ahora que ya hemos matado la tarea, podemos usar el comando `jobs` de nuevo para comprobarlo:

```
/home/larry# jobs

[1]+  Terminated              yes >/dev/null

/home/larry#
```

La tarea está, en efecto, muerta, y si usa el comando `jobs` de nuevo, no mostrará nada.

También podrá matar la tarea usando el número de ID de proceso (PID), el cual se muestra conjuntamente con el ID de tarea cuando arranca la misma. En nuestro ejemplo el ID de proceso es 164, así que el comando

```
/home/larry# kill 164
```

es equivalente a

```
/home/larry# kill %1
```

No es necesario usar el "`%`" cuando nos referimos a una tarea a través de su ID de proceso.

3.11.4 Parada y relanzamiento de tareas

Hay otra manera de poner una tarea en segundo plano. Usted puede lanzarlo como un proceso normal (en primer plano), pararlo, y después relanzarlo en segundo plano.

Primero, lance el proceso `yes` en primer plano como lo haría normalmente:

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

De nuevo, dado que `yes` corre en primer plano, no debe retornar el prompt de la shell.

Ahora, en vez de interrumpir la tarea con `ctrl-C`, *suspenderemos* la tarea. El suspender una tarea no la mata: solamente la detiene temporalmente hasta que Ud. la retoma. Para hacer esto usted debe pulsar la tecla de suspender, que suele ser `ctrl-Z`.

```
/home/larry# yes > /dev/null
ctrl-Z
[1]+  Stopped                 yes >/dev/null
/home/larry#
```

Mientras el proceso está suspendido, simplemente no se esta ejecutando. No gasta tiempo de CPU en la tarea. Sin embargo, usted puede retomar el proceso de nuevo como si nada hubiera pasado. Continuará ejecutandose donde se dejó.

Para relanzar la tarea en primer plano, use el comando `fg` (del inglés "foreground").

```
/home/larry# fg
yes >/dev/null
```

La shell muestra el nombre del comando de nuevo, de forma que tenga conocimiento de que tarea es la que ha puesto en primer plano. Pare la tarea de nuevo, con `ctrl-Z`. Esta vez utilice el comando `bg` para poner la tarea en segundo plano. Esto hará que el comando siga ejecutandose igual que si lo hubiese hecho desde el principio con "`&`" como en la sección anterior.

```
/home/larry# bg
[1]+ yes >/dev/null &
/home/larry#
```

Y tenemos de nuevo el prompt. El comando `jobs` debería decirnos que `yes` se está ejecutando, y podemos matar la tarea con `kill` tal y como lo hicimos antes.

¿Cómo podemos parar la tarea de nuevo? Si pulsa `ctrl-Z` no funcionará, ya que el proceso está en segundo plano. La respuesta es poner el proceso en primer plano de nuevo, con el comando `fg`, y entonces pararlo. Como puede observar podrá usar `fg` tanto con tareas detenidas, como con las que estén segundo plano.

Hay una gran diferencia entre una tarea que se encuentra en segundo plano, y una que se encuentra 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 (la tarea aun ocupa un lugar en memoria, aunque puede ser volcada a disco). Una tarea en segundo plano, se está ejecutando, y usando memoria, a la vez que completando alguna acción mientras usted hace otro trabajo. Sin embargo, una terea en segundo plano puede intentar mostrar texto en su terminal, lo que puede resultar molesto si está intentando hacer otra cosa. Por ejemplo, si usted usó el comando

```
/home/larry# yes &
```

sin redirigir stdout a `/dev/null`, una cadena de `y`-es se mostrarán en su monitor, sin modo alguno de interrumpirlo (no puede hacer uso de `ctrl-C` para interrumpir tareas en segundo plano). Para poder parar esas interminables `y`-es, tendría que usar el comando **fg** para pasar la tarea a primer plano, y entonces usar `ctrl-C` para matarla.

Otra observación. Normalmente, los comandos "fg" y "bg" actúan sobre el último proceso parado (indicado por un "+" junto al número de tarea cuando usa el comando `jobs`). Si usted tiene varios procesos corriendo a la vez, podrá mandar a primer o segundo plano una tarea específica indicando el ID de tarea como argumento de `fg` o `bg`, como en

```
/home/larry# fg %2
```

(para la tarea de primer plano número 2), o

```
/home/larry# bg %3
```

(para la tarea de segundo plano número 3). No se pueden usar los ID de proceso con `fg` o `bg`.

Además de esto, si usa el número de tarea por si solo, como

```
/home/larry# %2
```

es equivalente a

```
/home/larry# fg %2
```

Solo recordarle que el uso de control de tareas es una utilidad de la shell. Los comandos `fg`, `bg` y `jobs` son internos de la shell. Si por algún motivo usted utiliza una shell que no soporta control de tareas, no espere disponer de estos comandos.

Y además, hay algunos aspectos del control de tareas que difieren entre Bash y Tcsh. De hecho, algunas shells no proporcionan ningún control de tareas—sin embargo, la mayoría de las shells disponibles para Linux soportan control de tareas.

3.12 Usando el editor vi

Un **editor de texto** es simplemente un programa usado para la edición de ficheros que contienen texto, como una carta, un programa en C, o un fichero de configuración del sistema. Mientras que hay muchos editores de texto disponibles en Linux, el único editor que está garantizado encontrar en cualquier sistema UNIX es `vi`—el "visual editor". `vi` no es el editor más fácil de usar, ni es muy autoexplicativo. De cualquier forma, como es tan común en el mundo UNIX y es posible que alguna vez necesite usarlo, aquí encontrará algo de documentación.

La elección de un editor es principalmente una cuestión de gusto personal y estilo. Muchos usuarios prefieren el barroco, autoexplicativo y potente **Emacs**—un editor con más características que cualquier otro programa único en el mundo UNIX. Por ejemplo, Emacs tiene integrado su propio dialecto del lenguaje de programación LISP y tiene muchas extensiones (una de ellas es el programa "Eliza"- como programa de IA). Pero como Emacs y todos sus ficheros de soporte es relativamente grande, puede que no tenga acceso a él en muchos sistemas. `vi`, por otra parte, es pequeño y potente, pero más difícil de usar. De cualquier modo, una vez que conozca la forma de funcionamiento de `vi`, es muy fácil usarlo. Simplemente la curva de aprendizaje es bastante pronunciada al comienzo.

Esta sección es una introducción coherente a `vi`—no discutiremos todas sus características, solo aquellas necesarias para que sepa como comenzar. Puede dirigirse a la página de manual de `vi` si está interesado en aprender más acerca de las características de este editor, o puede leer el libro *Learning the* `vi` *Editor* de O'Reilly and Associates. Vea el Apéndice A para información.

3.12.1 Conceptos

Mientras se usa `vi`, en cualquier momento estará en uno de tres posibles modos de operación. Estos modos son conocidos como *modo órdenes*, *modo inserción* y *modo última línea*.

Cuando inicia `vi`, está en el *modo órdenes*. Este modo le permite usar ciertas órdenes para editar ficheros o cambiar a otros modos. Por ejemplo, tecleando "`x`" mientras está en el modo órdenes, borra el carácter que hay debajo del cursor. Las teclas del cursor mueven este por el fichero que estamos editando. Generalmente, las órdenes usadas en este modo son solo de uno o dos carácteres de longitud.

Habitualmente insertará o editará texto desde el *modo inserción*. Usando `vi`, probablemente dedicará la mayor parte del tiempo en este modo. Inicia el modo de inserción al usar una orden como "`i`" (para "insertar") desde el modo de órdenes. Una vez en el modo de inserción, irá insertando texto en el documento desde la posición actual del cursor. Para salir del modo de inserción y volver al de órdenes, pulse `esc`.

Modo última línea es un modo especial usado para proporcionar ciertas órdenes extendidas a `vi`. Al usar esos comandos, aparecen en la última línea de la pantalla (de ahí el nombre). Por ejemplo, cuando teclea "`:`" desde el modo de órdenes, entrará en el modo última línea, y podrá usar órdenes como "`wq`" (para escribir el fichero a disco y salir de `vi`), o "`q!`" (para salir de `vi` sin guardar los cambios). El modo de última línea es habitualmente usado por órdenes `vi` mayores de un carácter. En el modo de última línea, introduce una orden de una sola línea y pulsa `enter` para ejecutarla.

3.12.2 Comenzando con `vi`

La mejor forma de entender estos conceptos es arrancar `vi` y editar un fichero. En el ejemplo "screens" que veremos, vamos a mostrar solo unas pocas líneas de texto, como si la pantalla tuviese solo seis líneas de altura (en lugar de veinticuatro).

La sintaxis de `vi` es

`vi` ⟨*filename*⟩

donde ⟨*filename*⟩ es el nombre del fichero que desea editar.

Arranque `vi` tecleando

```
/home/larry# vi test
```

lo que editará el fichero `test`. Debería ver algo como

```
~
~
~
~
~
~
"test" [New file]
```

La columna de carácteres "~" indica que está al final del fichero.

3.12.3 Insertando texto

Está ahora en modo órdenes; para poder insertar texto en el fichero, pulse `i` (lo que le hará entrar en modo inserción), y comience a escribir.

```
Now is the time for all good men to come to the aid of the party._
~
~
~
~
~
```

Mientras inserta texto, puede escribir tantas líneas como desee (pulsando `return` después de cada una, por supuesto), y puede corregir los errores con la tecla de borrado de carácter.

Para salir del modo de inserción y volver al modo de órdenes, pulse `esc`.

Mientras esté en modo órdenes, puede usar las teclas del cursor para moverse por el fichero. En nuestro ejemplo, como solo tenemos una línea, el tratar de usar las teclas de línea arriba o abajo, probablemente hará que `vi` emita un pitido.

Hay muchas formas de insertar texto a parte de la orden `i`. Por ejemplo, la orden `a` inserta texto comenzando *detrás* de la posición actual del cursor, en lugar de la posición actual del cursor. Por ejemplo, use la tecla de cursor a la izquierda para desplazar el cursor entre las palabras "good" y "men".

```
Now is the time for all good_men to come to the aid of the party.
~
~
~
~
~
```

Pulse `a` para iniciar el modo inserción, teclee "`wo`" y pulse `esc` para volver al modo de órdenes.

```
Now is the time for all good women to come to the aid of the party.
~
~
~
~
~
```

Para comenzar a insertar texto en la línea de debajo de la actual, use la orden "o". Por ejemplo, pulse `o` y teclee otra línea o dos:

```
Now is the time for all good women to come to the aid of the party.
Afterwards, we'll go out for pizza and beer.
~
~
~
~
```

Solo recuerde que en cualquier momento está en modo de órdenes (donde órdenes como `i`, `a` o `o` son válidas, o en modo de inserción (cuando esté insertando texto, pulse `esc` para volver al modo de órdenes), o en modo de última línea (donde puede introducir comandos extendidos, como veremos más adelante).

3.12.4 Borrando texto

Desde el modo de órdenes, la orden `x` borra el carácter debajo del cursor. Si pulsa `x` cinco veces, terminará con:

```
Now is the time for all good women to come to the aid of the party.
Afterwards, we'll go out for pizza and_
~
~
~
~
```

Ahora pulse `a`, inserte algún texto, seguido de `esc`:

```
Now is the time for all good women to come to the aid of the party.
Afterwards, we'll go out for pizza and Diet Coke.
~
~
~
~
```

Puede borrar líneas enteras usando la orden `dd` (es decir, pulse `d` dos veces en una fila). Si el cursor está en la segunda línea y teclea `dd`,

```
Now is the time for all good women to come to the aid of the party.
~
~
~
~
~
```

Para borrar la palabra sobre la que se encuentra el cursor, use la orden `dw`. Situe el cursor sobre la palabra “good” y pulse `dw`.

```
Now is the time for all women to come to the aid of the party.
~
~
~
~
~
```

3.12.5 Modificando texto

Puede sustituir secciones de texto usando la orden `R`. Situe el cursor en la primera letra de “party” y pulse `R`, y escriba la palabra “hungry”.

```
Now is the time for all women to come to the aid of the hungry.
~
~
~
~
~
```

El uso de `R` para editar texto es bastante parecido al uso de las órdenes `i` y `a`, pero `R` sobreescribe texto en lugar de insertarlo.

La orden `r` sustituye un único carácter situado debajo del cursor. Por ejemplo, situe el cursor al comienzo de la palabra “Now” y escriba `r` seguido de `C`. Obtendrá:

```
Cow is the time for all women to come to the aid of the hungry.
~
~
~
~
~
```

La orden “`~`” cambia de mayúsculas a minúsculas o viceversa la letra sobre la que se encuentra el cursor. Por ejemplo, si situa el cursor sobre la “o” de “Cow”, y repetidamente pulsa `~`, obtendrá:

```
COW IS THE TIME FOR ALL WOMEN TO COME TO THE AID OF THE HUNGRY._
~
~
~
~
~
```

3.12.6 Ordenes de movimiento

Ya conoce como usar las teclas del cursor para moverse por el documento. Además, puede usar las órdenes `h`, `j`, `k` y `l` para mover el cursor a la izquierda, abajo, arriba y derecha respectivamente. Esto es muy cómodo cuando (por alguna razón) sus teclas de cursor no funcionen correctamente.

La orden `w` mueve el cursor al comienzo de la siguente palabra; `b` lo lleva al comienzo de la palabra anterior.

La orden `0` (cero) mueve el cursor al comienzo de la línea actual, y la orden `$` lo lleva al final de la línea.

Al editar ficheros grandes, querrá moverse hacia adelante y atrás a lo largo del fichero mostrando una pantalla cada vez. Pulsando `ctrl-F` avanza el cursor una pantalla hacia adelante y `ctrl-B` lo lleva una pantalla atrás.

Para llevar el cursor al final del fichero, pulse `G`. Puede también desplazarse a una línea arbitraria; por ejemplo, pulsando la orden `10G` llevará el cursor a la línea 10 del fichero. Para desplazarse al comienzo, use `1G`.

Puede asociar órdenes de desplazamiento con otras órdenes como es el borrado. Por ejemplo, la orden `d$` borrará todo desde la posición del cursor al final de la línea; `dG` borrará todo desde la posición del cursor al final del fichero.

3.12.7 Guardando ficheros y saliendo de `vi`

Para salir de `vi` sin modificar el fichero use la orden `:q!`. Al pulsar "`:`", el cursor se desplazará a la última línea de la pantalla; está en modo última línea.

```
COW IS THE TIME FOR ALL WOMEN TO COME TO THE AID OF THE HUNGRY.
~
~
~
~
~
:_
```

En el modo de última línea hay disponibles una serie de órdenes extendidas. Una de ellas es `q!`, la cual permite salir de `vi` sin guardar los cambios. La orden `:wq` salva el fichero y sale de `vi`. La orden `ZZ` (desde el modo de órdenes, sin "`:`") es equivalente a `:wq`. Recuerde que debe pulsar `enter` después de introducir la orden para que esta se ejecute en el modo última línea.

Para salvar el fichero sin salir de vi, simplemente use `:w`.

3.12.8 Editando otro fichero

Para editar otro fichero use la orden `:e`. Por ejemplo, para dejar de editar el fichero `test` y en su lugar editar el fichero `foo`, use la orden

```
COW IS THE TIME FOR ALL WOMEN TO COME TO THE AID OF THE HUNGRY.
~
~
~
~
~
:e foo_
```

Si usa `:e` sin salvar primero el fichero, obtendrá el mensaje de error

```
No write since last change (":edit!"  overrides)
```

lo cual significa que `vi` no quiere editar otro fichero hasta que salve el primero. En este punto, puede usar `:w` para guardar el fichero original, y entonces usar `:e`, o puede usar la orden

```
COW IS THE TIME FOR ALL WOMEN TO COME TO THE AID OF THE HUNGRY.
~
~
~
~
~
:e!  foo_
```

El signo "!" le dice a `vi` lo que realmente desea usted—editar el nuevo fichero sin salvar los cambios del primero.

3.12.9 Incluyendo otros ficheros

Si usas la orden `:r` Puede incluir el contenido de otro fichero en el fichero que está editando. Por ejemplo

```
:r foo.txt
```

insertaría el contenido del fichero `foo.txt` en el texto en la posición actual de cursor.

3.12.10 Ejecutando comandos del intérprete

Puede también ejecutar comandos del intérprete desde el interior de `vi`. La orden `:r!` funciona como `:r`, pero en lugar de leer un fichero, inserta la salida de un comando dado en el fichero en la posición actual del cursor. Por ejemplo, si usa la orden

```
:r! ls -F
```

obtendrá

```
COW IS THE TIME FOR ALL WOMEN TO COME TO THE AID OF THE HUNGRY.
letters/
misc/
papers/
~
~
```

También puede salir a un intérprete de comandos desde `vi`, es decir, ejecutar una orden desde dentro de `vi` y volver al editor una vez esta finalize. Por ejemplo, si usa la orden

```
:! ls -F
```

la orden `ls -F` será ejecutada, y los resultados mostrados en la pantalla, pero no insertados en el fichero en edición. Si usa la orden

```
:shell
```

`vi` iniciará una instancia del intérprete de comandos, permitiendole temporalmente dejar a vi "parado" mientras ejecuta otras órdenes. Simplemente salga del intérprete de comandos (usando la orden `exit`) para regresar a `vi`

3.12.11 Obteniendo ayuda

`vi` no proporciona demasiada ayuda de forma interactiva (la mayoría de los programas UNIX no lo hacen), pero siempre puede leer la página de manual para `vi`. `vi` es un "front-end" visual para el editor `ex`: es decir, es `ex` quien maneja la mayoría de las órdenes en el modo última línea. Luego además de leer la página de `vi`, consulte la de `ex` también.

3.13 Personalizando su entorno

El intérprete de comandos proporciona muchos mecanismos para personalizar su entorno de trabajo. Como hemos mencionado antes, el intérprete de comandos es más que un mero intérprete—es también un poderoso lenguaje de programación. Aunque escribir guiones del intérprete de comandos es una tarea extensa, nos gustaría introducirle algunas formas en las que puede simplificar su trabajo en un sistema UNIX mediante el uso de características avanzadas del intérprete.

Como mencionamos antes, diferentes intérpretes usan diferentes sintaxis para la ejecución de guiones. Por ejemplo, Tcsh usa una notación al estilo C, mientras que Bourne usa otro tipo de sintaxis. En esta sección no nos fijaremos en las diferencias entre los dos y supondremos que los guiones son escritos con la sintaxis del intérprete de comandos Bourne.

3.13.1 Guiones del intérprete de comandos

Supongamos que usa una serie de comandos a menudo, y le gustaría acortar el tiempo requerido para teclear agrupándolos en una única "orden". Por ejemplo, las órdenes

```
/home/larry# cat chapter1 chapter2 chapter3 > book
/home/larry# wc -l book
/home/larry# lp book
```

concatenarán los ficheros `chapter1`, `chapter2` y `chapter3` y guardará el resultado en el fichero `book`. Entonces, se mostrará el recuento del número de líneas del fichero `book` y finalmente se imprimirá con el comando `lp`.

En lugar de teclear todos esos comandos, podría agruparlos en un **guión del intérprete de comandos**. Describimos los guiones brevemente en la Sección 3.13.1. El guión usado para ejecutar todas las órdenes sería

```
#!/bin/sh
# A shell script to create and print the book

cat chapter1 chapter2 chapter3 > book
wc -l book
lp book
```

Si el guión se salva en el fichero `makebook`, podría simplemente usar la orden

```
/home/larry# makebook
```

para ejecutar todas las órdenes del guión. Los guiones son simples ficheros de texto; puede crearlos con un editor como `emacs` o `vi` [8].

Veamos este guión. La primera línea "`#!/bin/sh`", identifica el fichero como un guión y le dice al intérprete de comandos como ejecutarlo. Instruye al intérprete a pasarle el guión a `/bin/sh` para la ejecución, donde `/bin/sh` es el programa del intérprete. ¿Por qué es esto importante? En la mayoría de los sistemas UNIX `/bin/sh` es un intérprete de comandos Bourne, como Bash. Forzando al guión a ejecutarse usando `/bin/sh` nos estamos asegurando de que será interpretado según la sintaxis de Bourne. Esto hará que el guión se ejecute usando la sintaxis Bourne aunque esté usando Tcsh como intérprete de comandos.

La segunda línea es un *comentario*. Estos comienzan con el carácter "`#`" y continuan hasta el final de la línea. Los comentarios son ignorados por el intérprete de comandos—son habitualmente usados para identificar el guión con el programador.

El resto de las líneas del guión son simplemente órdenes como las que podría teclear directamente. En efecto, el intérprete de comandos lee cada línea del guión y ejecuta la línea como si hubiese sido tecleada en la línea de comandos.

[8] `vi` se describe en la Sección 3.12.

Los permisos son importantes para los guiones. Si crea un guión, debe asegurarse de que tiene permisos de ejecución para poder ejecutarlo[9]. La orden

`/home/larry#` *chmod u+x makebook*

puede ser usada para dar permisos de ejecución al guión `makebook`.

3.13.2 Variables del intérprete de comandos y el entorno

El intérprete de comandos le permite definir **variables** como la mayoría de los lenguajes de programación. Una variable es simplemente un trozo de datos al que se le da un nombre.

◇ Notese que Tcsh, así como otros intérpretes del estilo C, usan un mecanismo diferente para inicializar variables del descrito aquí. Esta discusión supondrá el uso del intérprete Bourne, como es Bash (el cual probablemente está usando). Vea la página de manual de Tcsh para más detalles.

Cuando asigna un valor a una variable (usando el operador "="), puede acceder a la variable añadiendo a su nombre "$", como se ve a continuación.

`/home/larry#` *foo="hello there"*

A la variable `foo` se le da el valor "`hello there`". Podemos ahora hacer referencia a ese valor a través del nombre de la variable con el prefijo "`$`". La orden

```
/home/larry# echo $foo
hello there
/home/larry#
```

produce el mismo resultado que

```
/home/larry# echo "hello there"
hello there
/home/larry#
```

Estas variables son internas al intérprete. Esto significa que solo éste podrá acceder a las variables. Esto puede ser útil en los guiones; si necesita mantener, por ejemplo, el nombre de un fichero, puede almacenarlo en una variable. Usando la orden `set` mostrará una lista de todas las variables definidas en el intérprete de comandos.

De cualquier modo, el intérprete de comandos permite **exportar** variables al **entorno**. El entorno es el conjunto de variables a las cuales tienen acceso todas las órdenes que ejecute. Una vez que se define una variable en el intérprete, exportarla hace que se convierta también en parte del entorno. La orden `export` es usada para exportar variables al entorno.

◇ De nuevo, hemos de diferenciar entre Bash y Tcsh. Si está usando Tcsh, deberá usar una sintaxis diferente para las variables de entorno (se usa la orden `setenv`). Diríjase a la página de manual de Tcsh para más información.

[9] Cuando crea ficheros de texto, los permisos por defecto usualmente no incluyen los de ejecución.

El entorno es muy importante en un sistema UNIX. Le permite configurar ciertas órdenes simplemente inicializando variables con las órdenes ya conocidas.

Veamos un ejemplo rápido. La variable de entorno `PAGER` es usada por la orden `man`. Especifica la orden que se usará para mostrar las páginas del manual una a una. Si inicializa `PAGER` con el nombre del programa, se usará este para mostrar las páginas de manual en lugar de `more` (el cual es usado por defecto).

Inicialice `PAGER` a "`cat`". Esto hará que la salida de `man` sea mostrada de una, sin pausas entre páginas.

```
/home/larry# PAGER="cat"
```

Ahora exportamos `PAGER` al entorno.

```
/home/larry# export PAGER
```

Puebe la orden `man ls`. La página debería volar por su pantalla sin detenerse entre páginas.

Ahora, si inicializa `PAGER` a "`more`", se usará la orden `more` para mostrar las páginas del manual.

```
/home/larry# PAGER="more"
```

Nótese que no hemos de usar la orden `export` después del cambio de la variable `PAGER`. Solo hemos de exportar las variables una vez; cuaquier cambio efectuado con posterioridad será automáticamente propagado al entorno.

Las páginas de manual para una orden en particular, le informarán acerca del uso de alguna variable de entorno por parte de esa orden; por ejemplo, la página de manual de `man` explica que `PAGER` es usado para especificar la orden de paginado.

Algunas órdenes comparten variables de entorno; por ejemplo, muchas órdenes usan la variable `EDITOR` para especificar el editor por defecto para usar si es necesario.

El entorno es también usado para guardar información importante acerca de la sesión en curso. Un ejemplo es la variable de entorno `HOME`, que contiene el nombre del directorio de origen del usuario.

```
/home/larry/papers# echo $HOME
/home/larry
```

Otra variable de entorno interesante es `PS1`, la cual define el "prompt" principal que usará el intérprete. Por ejemplo,

```
/home/larry# PS1="Your command, please: "
Your command, please:
```

Para volver a inicializar el "prompt" a su valor habitual (el cual contiene el directorio actual seguido por el símbolo "`#`"),

```
Your command, please:   PS1="\w# "
/home/larry#
```

La página de manual de `bash` describe la sintaxis usada para inicializar el "prompt".

3.13.2.1 La variable de entorno PATH

Cuando usa la orden `ls` ¿cómo encuentra el intérprete el programa ejecutable `ls`?. De hecho, `ls` se encuentra en `/bin/ls` en la mayoría de los sistemas. El intérprete usa la variable de entorno `PATH` para localizar los ficheros ejecutables u órdenes que tecleamos.

Por ejemplo, su variable `PATH` puede inicializarse a:

```
/bin:/usr/bin:/usr/local/bin:.
```

Esto es una lista de directorios en los que el intérprere debe buscar. Cada directorio está separado por un "`:`". Cuando usa la orden `ls`, el intérprete primero busca `/bin/ls`, luego `/usr/bin/ls` y así hasta que lo localice o acabe la lista.

Nótese que `PATH` no interviene en la localización de ficheros regulares. Por ejemplo, si usa la orden

```
/home/larry# cp foo bar
```

El intérprete no usará `PATH` para localizar los ficheros `foo` y `bar`—esos nombres se suponen completos. Solo se usará `PATH` para localizar el programa ejecutable `cp`.

Esto le permitirá ahorrar mucho tiempo; significa que no deberá recordar donde son guardadas las órdenes. En muchos sistemas los ficheros ejecutables se dispersan por muchos sitios, como `/usr/bin`, `/bin` o `/usr/local/bin`. En lugar de dar el nombre completo con el camino (como `/usr/bin/cp`), solo hemos de inicializar `PATH` con la lista de los directorios donde queremos que se busquen automáticamente.

Nótese que `PATH` contiene "`.`", el cual es el directorio actual de trabajo. Esto le permite crear guiones o programas y ejecutarlos desde su directorio de trabajo actual sin tener que especificarlo directamente (como en `./makebool`). Si un directorio no está en su `PATH`, entonces el intérprete no buscará en él ordenes para ejecutar—esto incluye al directorio de trabajo.

3.13.3 Guiones de inicialización del intérprete

A parte de los guiones que puede crear, hay un número de estos que usa el intérprete de comandos para ciertos propósitos. Los más importantes son sus **guiones de inicialización**, guiones automáticamente ejecutados por el intérprete al abrir una sesión.

Los guiones de inicialización son eso, simples guiones como los descritos arriba. De cualquier modo, son muy útiles para la inicialización de su entorno al ejecutarse automáticamente. Por ejemplo, si siempre usa la orden `mail` para comprobar si tiene correo al iniciar una sesión, incluya en su guión de inicialización dicha orden y será ejecutada automáticamente.

Tanto Bash como Tcsh distinguen entre un **intérprete de presentación** y otras invocaciones del intérprete. Un intérprete de presentación es el que se ejecuta en el momento de la presentación al sistema (login). Es el único que usará. De cualquier modo, si ejecuta una opción de salir a un intérprete desde algún programa, como `vi`, inicializa otra instancia del intérprete de comandos, el cual no es su intérprete de presentación. Además, en cualquier momento que ejecuta un guión, automáticamente está arrancando otro intérprete que va a ser el encargado de ejecutar el guión.

Los ficheros de inicialización usados por Bash son: `/etc/profile` (configurado por el administrador del sistema, y ejecutado por todos los usuarios de Bash en el momento de la presentación al sistema), `$HOME/.bash_profile` (ejecutado por una sesión de presentación Bash) y `$HOME/.bashrc` (ejecutadas por todas las sesiones Bash que no son de presentación). Si `.bash_profile` no está presente, se usa en su lugar `.profile`

Tcsh usa los siguientes guiones de inicialización: `/etc/csh.login` (ejecutado por todos los usuarios de Tcsh en el momento de la presentación al sistema), `$HOME/.tcshrc` (ejecutado en la presentación al sistema por todas las instancias nuevas de Tcsh) y `$HOME/.login` (ejecutado en la presentación al sistema, seguido `.tcshrc`). Si `.tcshrc` no está presente, `.cshrc` se usa en su lugar.

Para entender completamente la función de estos ficheros, necesitará aprender más acerca del intérprete de comandos. La programación de guiones es una materia complicada, más allá del alcance de este libro. Lea las páginas de manual de `bash` y/o `tcsh` para aprender más sobre la configuración de su entorno.

3.14 ¿Quieres seguir por tu cuenta?

Esperamos haberle proporcionado suficiente información para darle una idea básica de como usar el sistema, teniendo en cuenta que la mayoría de los aspectos más importantes e interesantes de Linux no están cubiertos aquí— esto es muy básico. Con esta base, en poco tiempo estará ejecutando complicadas aplicaciones y aprovechando todo el potencial de tu sistema. Si la cosa no es muy excitante al comienzo, no desespere— hay mucho que aprender.

Una herramienta indispensable para aprender acerca del sistema son las páginas del manual. Aunque muchas de las páginas pueden parecer confusas al principio, si se profundiza hay gran cantidad de información en ellas.

También es interesante leer un libro sobre la utilización de un sistema UNIX. Hay mucho más en UNIX de lo que pueda parecer a simple vista– desafortunadamente, la mayoría de ello queda fuera del alcance de este libro. En el apéndice A encontrarás una lista de algunos buenos libros sobre UNIX.

Capítulo 4

Administración del Sistema

Este capítulo es una visión general de la administración de un sistema Linux, incluyendo un número de posibilidades avanzadas que no son, necesariamente, sólo para administradores de sistemas. Igual que cada perro tiene su día, cada sistema tiene su administrador, y poner en marcha el sistema es un trabajo muy importante y a veces consume mucho tiempo, incluso si se es el único usuario en el sistema.

Hemos intentado cubrir aquí los elementos más importantes acerca de la administración de sistemas que se necesitan conocer cuando se use Linux, en suficiente detalle para empezar confortablemente. Para mantenerlo corto y agradable, sólo hemos cubierto los niveles más básicos y nos hemos saltado muchos e importantes detalles. Se debe leer el *Linux System Administrator's Guide* si se quiere ejecutar Linux en serio. Le ayudará a comprender mejor como funcionan las cosas y como se ensamblan juntas. Al menos hojéelo para que pueda conocer qué contiene y qué tipo de ayuda puede esperar de él.

4.1 Acerca de Raíces, Sombreros y la Sensación de Poder

Como sabe, UNIX distingue entre diferentes usuarios para que lo que hagan a los demás y al sistema pueda ser regulado (uno no desearía que nadie pudiese leer nuestras cartas de amor, por ejemplo). Cada usuario recibe una **cuenta** que incluye un nombre de usuario, un directorio inicial, y otras cosas por el estilo. Además de las cuentas dadas a personas reales, existen cuentas especiales, definidas por el sistema, que tienen privilegios especiales. La más importante de éstas es la **cuenta raíz**, con el nombre de usuario `root`.

4.1.1 La cuenta `root`

Los usuarios normales están restringidos normalmente para que no puedan dañar a nadie más en el sistema, sólo a ellos mismos. Los permisos de los ficheros en el sistema están preparados para que los usuarios normales no tengan permitido borrar o modificar ficheros en directorios compartidos por todos los usuarios (como son /bin y /usr/bin. Muchos usuarios también protegen sus propios ficheros con los permisos adecuados para que otros usuarios no puedan acceder o modificar éstos ficheros.

Estas restricciones desaparecen para root. El usuario root puede leer, modificar o borrar cualquier fichero en el sistema, cambiar permisos y pertenencias en cualquier fichero, y ejecutar programas especiales, como pueden ser los que particionan un disco o crean sistemas de ficheros. La idea básica es que la persona o personas que ejecutan y cuidan del sistema entren como root cuando sea necesario para realizar tareas que no pueden ser ejecutadas por un usuario normal. Puesto que root puede hacer todo, es fácil cometer errores que tengan consecuencias catastróficas cuando se trabaja utilizando esta cuenta.

Por ejemplo, como un usuario normal, si inadvertidamente se intentase borrar todos los ficheros en /etc, el sistema no lo permitiría. Sin embargo, como usuario root, el sistema no diría nada. Es muy simple el dañar el sistema utilizando root. La mejor forma de evitar accidentes es:

- Pensárselo dos veces antes de apretar return en un comando que pueda causar daño. Por ejemplo, si se va a borrar un directorio, antes de pulsar return, releer el comando completo y asegurarse que es correcto.

- No acostumbrarse a utilizar root. Cuanto más confortable se encuentre uno trabajando con el usuario root, más seguro que se confundirán los privilegios con los de un usuario normal. Por ejemplo, puede uno *pensar* que se está conectado como larry, cuando realmente se está conectado como root.

- Utilice un marcador distinto para la cuenta root. Se debe cambiar el fichero .bashrc o el .login de root para poner el marcador del intérprete a algo distinto del marcador de usuario normal. Por ejemplo, mucha gente utiliza el carácter "$" como marcador para los usuarios normales, y reserva el carácter "#" como marcador para el usuario root.

- Conectarse como root sólo cuando sea absolutamente necesario. Y desconectarse tan pronto como se haya terminado el trabajo. Cuanto menos se use la cuenta root, menos posibilidades habrá de dañar el sistema.

Por supuesto, existe una variante de hackers de UNIX que utilizan root para prácticamente todo. Pero cada uno de ellos ha cometido, en algún momento, algún error tonto como root y corrompido el sistema. La regla general es que hasta que se familiarice con la falta de restricciones de root, y esté cómodo utilizando el sistema sin dichas restricciones, procure conectarse como root lo menos posible.

Por supuesto, todo el mundo comete errores. El mismo Linus Torvalds borró accidentalmente una vez el árbol de directorios completo del núcleo de su sistema. Horas de trabajo se perdieron para siempre. Afortunadamente, sin embargo, gracias a su conocimiento del código del sistema de ficheros, fue capaz de reiniciar el sistema y reconstruir el árbol de directorios del disco a mano.

Pongámoslo de otra forma, si se imagina la utilización de la cuenta root como la utilización de un sombrero mágico especial que le da montones de poderes, pudiendo, con un gesto de la mano, destruir ciudades enteras, es una buena idea de lo que es ser un poco cauto con lo que se hace con las manos. Puesto que es fácil mover la mano de manera destructiva por accidente, no es una buena idea el llevar el sombrero mágico cuando no es necesario, a pesar de la maravillosa sensación.

4.1.2 Abusando del sistema

Junto con el sentimiento de poder llega la tendencia a hacer daño. Este es uno de los puntos oscuros de la administración de sistemas UNIX, pero todo el mundo pasa por ello en algún momento. Muchos usuarios de sistemas UNIX nunca tienen la posibilidad de manejar este poder —en los sistemas UNIX de universidad y de empresas, solo los altamente cualificados (y altamente pagados) administradores de sistemas llegan a conectarse como `root`. De hecho, en muchas de esas instituciones, la clave de `root` es un secreto celosamente guardado: es tratado como el Sagrado Grial de la institución. Se monta mucho jaleo con la conexión como `root`; está visto como un sabio y a la vez espantoso poder, dado sólo a un reducido grupo de elegidos.

Este tipo de actitud hacia la cuenta `root` es, sencillamente, el tipo de actitud que alimenta la malicia y el desprecio. Ya que `root` es tan atractivo, cuando algunos usuarios tienen su primera oportunidad de conectarse como `root` (ya sea en un sistema Linux, o en cualquier otro sitio), la tendencia es a utilizar los privilegios de `root` de forma descuidada. Yo he conocido "administradores de sistemas" (por llamarlos de alguna forma) que leen el correo de otros usuarios, borran ficheros de usuario sin avisar y que, de forma general, se comportan como niños cuando se les da un "juguete" poderoso.

Puesto que `root` tiene tantos privilegios en el sistema, se necesita una cierta madurez y autocontrol para utilizar la cuenta de la forma para la que está diseñada —para ejecutar el sistema. Existe un código de honor no hablado entre el administrador y los usuarios del sistema. ¿Cómo se sentiría si su administrador de sistemas leyese su correo electrónico o mirase sus ficheros? Aún no hay ningún precedente legal firme acerca de la intimidad electrónica en sistemas de ordenadores de tiempo compartido. En sistemas UNIX, el usuario `root` tiene la posibilidad de saltarse todos los mecanismos de seguridad y privacidad del sistema. Es importante que el administrador del sistema desarrolle una relación de confianza con los usuarios del sistema. Es algo en lo que nunca se puede insistir lo suficiente.

4.1.3 Como proceder con los usuarios

La seguridad en UNIX es bastante laxa por diseño. La seguridad del sistema fue un pensamiento posterior —el sistema se desarrolló originalmente en un entorno donde las intrusiones de un usuario en otro eran algo inaudito. Por esto, incluso con medidas de seguridad, aún existe la posibilidad de que usuarios normales hagan daño.

Los administradores de sistemas pueden tomar dos posturas cuando traten con usuarios abusivos: ser paranoicos o confiados. El administrador de sistemas paranoico normalmente causa más daño que el que previene. Una de mis citas favoritas es, "Nunca atribuyas a la malicia nada que pueda ser atribuido a la estupidez." Dicho de otra forma, muchos usuarios no tienen la habilidad o el conocimiento para hacer daño real al sistema. El 90% del tiempo, cuando un usuario causa problemas en el sistema (por ejemplo, rellenando la partición de usuarios con grandes ficheros, o ejecutando múltiples instancias de un gran programa), el usuario simplemente desconoce que lo que está haciendo es un problema. He ido a ver a usuarios que estaban causando una gran cantidad de problemas, pero su actitud estaba causada por la ignorancia —no por la malicia.

Cuando se enfrente con usuarios que puedan causar problemas potenciales no sea acusativo. La antigua regla de "inocente hasta que se demuestre lo contrario" sigue siendo válida. Es mejor una simple charla con el usuario, preguntándole acerca del problema, en lugar de causar una confrontación. Lo último que se desea es estar entre los malos desde el punto de vista del usuario. Esto levantaría un montón de sospechas acerca de si usted —el administrador de sistemas— tiene el sistema correctamente configurado. Si un usuario cree que usted le disgusta o no le tiene confianza, le puede acusar de borrar ficheros o romper la privacidad del sistema. Esta no es, ciertamente, el tipo de situación en la que quisiera estar.

Si encontrase que un usuario ha estado intentando "romper" el sistema, o ha estado haciendo daño al sistema de forma intencionada, no devuelva el comportamiento malicioso a su vez. En vez de ello, simplemente, dele un aviso —pero sea flexible. En muchos casos, se puede cazar a un usuario "con las manos en la masa" dañando al sistema —avísele. Dígale que no lo vuelva a repetir. Sin embargo, si le *vuelve* a cazar haciendo daño, entonces puede estar absolutamente seguro de que es intencionado. Ni siquiera puedo empezar a describir la cantidad de veces que parecía que había un usuario causando problemas al sistema, cuando de hecho, era o un accidente o un fallo mío.

4.1.4 Fijando las reglas

La mejor forma de administrar un sistema no es con un puño de hierro. Así puede ser como se haga lo militar, pero UNIX no fue diseñado para ese tipo de disciplina. Tiene sentido el escribir un conjunto sencillo y flexible de reglas para los usuarios —pero recuerde, cuantas menos reglas tenga, menos posibilidades habrá de romperlas. Incluso si sus reglas para utilizar el sistema son perfectamente razonables y claras, siempre habrá momentos en que los usuarios romperán dichas reglas sin pretenderlo. Esto es especialmente cierto en el caso de usuarios UNIX nuevos, que están aprendiendo los entresijos del sistema. No esta suficientemente claro, por ejemplo, que uno no debe bajarse un gigabyte de ficheros y enviárselo por correo a todos los usuarios del sistema. Los usuarios necesitan comprender las reglas y por que están establecidas.

Si especifica reglas de uso para su sistema, asegúrese de que el motivo detrás de cada regla particular esté claro. Si no lo hace, los usuarios encontrarán toda clase de formas creativas de saltársela y no saber que en realidad la están rompiendo.

4.1.5 Lo que todo esto significa

No podemos decirle como ejecutar su sistema al último detalle. Mucha de la filosofía depende en como se use el sistema. Si se tienen muchos usuarios, las cosas son muy diferentes de si solo tiene unos pocos o si se es el único usuario del sistema. Sin embargo, siempre es una buena idea —en cualquier situación— comprender lo que ser administrador de sistema significa en realidad.

Ser el administrador de un sistema no le hace a uno un mago del UNIX. Hay muchos administradores de sistemas que conocen muy poco acerca de UNIX. Igualmente, hay muchos usuarios "normales" que saben más acerca de UNIX que lo que cualquier administrador de sistemas. También, ser el administrador de sistemas no le permite el utilizar la malicia contra sus usuarios. Aunque el sistema le dé el privilegio de enredar en los ficheros de los usuarios, no significa que se tenga ningún derecho a hacerlo.

Por último, ser el administrador del sistema no es realmente una gran cosa. No importa si sus sistema es un pequeño 386 o un super ordenador Cray. La ejecución del sistema es la misma. El saber la clave de `root` no significa ganar dinero o fama. Tan solo le permitirá ejecutar el sistema y mantenerlo funcionando. Eso es todo.

4.2 Arrancando el Sistema

Hay varias maneras de arrancar el sistema, bien sea desde disquete o bien desde el disco duro.

4.2.1 Utilizando un disquete de arranque

Mucha gente arranca Linux utilizando un "disquete de arranque" que contiene una copia del núcleo de Linux. Este núcleo tiene la partición raíz de Linux codificada en él, para que sepa donde buscar en el disco duro el sistema de ficheros raíz. (El comando `rdev` puede ser utilizado para poner la partición raíz en la imagen del núcleo; ver más adelante.) Por ejemplo, este es el tipo de disquete creado por Slackware durante la instalación.

Para crear su propio disquete de arranque, localice en primer lugar la imagen del núcleo en su disco duro. Debe estar en el fichero `/Image` o `/etc/Image`. Algunas instalaciones utilizan el fichero `/vmlinux` para el núcleo.

En su lugar, puede que tenga un núcleo comprimido. Un núcleo comprimido se descomprime a sí mismo en memoria en tiempo de arranque, y utiliza mucho menos espacio en el disco duro. Si se tiene un núcleo comprimido, puede encontrarse en el fichero `/zImage` o `/etc/zImage`. Algunas instalaciones utilizan el fichero `/vmlinuz` para el núcleo comprimido.

Una vez que se sabe donde está el núcleo, hay que poner el nombre de la partición raíz de un dispositivo raíz en la imagen del núcleo, utilizando el comando `rdev`. El formato de este comando es

`rdev` ⟨*nombre-de-núcleo*⟩ ⟨*dispositivo-raíz*⟩

donde ⟨*nombre-del-núcleo*⟩ es el nombre de la imagen del núcleo, y ⟨*dispositivo-raíz*⟩ es el nombre de la partición raíz de Linux. Por ejemplo, para hacer que el dispositivo raíz en el núcleo `/etc/Image` sea `/dev/hda2`, utilice el comando

`#` *rdev /etc/Image /dev/hda2*

`rdev` también puede poner otras opciones en el núcleo, como puede ser el modo SVGA por defecto a utilizar en tiempo de arranque. Tan sólo utilice "`rdev -h`" para obtener un mensaje de ayuda.

Una vez puesto el dispositivo raíz, tan sólo hay que copiar la imagen del núcleo al disquete. Siempre que se copia datos a un disquete, es una buena idea formatear previamente el disquete en MS-DOS. Esto establece la información de pista y sector en el disquete con la que puede detectarse como de alta o baja densidad.

Por ejemplo, para copiar el núcleo en el fichero `/etc/Image` al disquete en `/dev/fd0`, se puede utilizar el comando

```
# cp /etc/Image /dev/fd0
```

Este disquete debe arrancar ahora Linux.

4.2.2 Utilizando LILO

Otro método de arranque es utilizar LILO, un programa que reside en el sector de arranque del disco duro. Este programa se ejecuta cuando el sistema se inicia desde el disco duro, y puede arrancar automáticamente Linux desde una imagen de núcleo almacenada en el propio disco duro.

LILO puede utilizarse también como una primera etapa de carga de varios sistemas operativos, permitiendo seleccionar en tiempo de arranque qué sistema operativo (como Linux o MS-DOS) arrancar. Cuando se arranca utilizando LILO, se inicia el sistema operativo por defecto, a menos que pulse `ctrl`, `alt`, o `shift` durante la secuencia de arranque. Si se pulsa cualquiera de estas teclas, se le presentará un indicador de arranque, donde debe teclear el nombre del sistema operativo a arrancar (como puede ser "`linux`" o "`msdos`"). Si se pulsa la tecla `tab` en el indicador de arranque, se le presentará una lista de los sistemas operativos disponibles.

La forma más simple de instalar LILO es editar el fichero de configuración, `/etc/lilo.conf`, y ejecutar el comando

```
# /sbin/lilo
```

El fichero de configuración de LILO contiene una "estrofa" para cada sistema operativo que se pueda querer arrancar. La mejor forma de mostrarlo es con un ejemplo de un fichero de configuración LILO. El ejemplo siguiente es para un sistema que tiene una partición raíz Linux en `/dev/hda1` y una partición MS-DOS en `/dev/hda2`.

```
# Le indicamos a LILO que modifique el registro de arranque de
# /dev/hda (el primer disco duro no-SCSI). Si se quiere arrancar desde
# una unidad distinta de /dev/hda, se debe cambiar la siguiente linea
boot = /dev/hda

# Nombre del cargador de arranque. No hay razon para cambiarlo, a menos
# que se este haciendo una modificacion seria del LILO
install = /boot/boot.b

# Dejemos a LILO efectuar alguna optimizacion.
compact
```

```
# Estrofa para la particion raiz de Linux en /dev/hda1.
image = /etc/Image   # Ubicacion del kernel
   label = linux     # Nombre del SO (para el menu de arranque de LILO)
   root = /dev/hda1  # Ubicacion de la particion raiz
   vga = ask         # Indicar al nucleo que pregunte por modos SVGA
                     #   en tiempo de arranque

# Estrofa para la particion MSDOS en /dev/hda2.
other = /dev/hda2    # Ubicacion de la particion
   table = /dev/hda  # Ubicacion de la tabla de particion para /dev/hda2
   label = msdos     # Nombre del SO (para el menu de arranque)
```

La primera “estrofa” de sistema operativo en el menú del fichero de configuración será el sistema operativo que arrancará LILO por defecto. Se puede seleccionar otro sistema operativo en el indicador de arranque de LILO, tal y como se indicó anteriormente.

Recuerde que cada vez que actualice la imagen del núcleo en disco, se debe reejecutar `/sbin/lilo` para que los cambios queden reflejados en el sector de arranque de su unidad.

También tenga en cuenta que si utiliza la lí nea “`root =`”, no hay motivo para utilizar `rdev` para poner la partición raíz en la imagen del núcleo. LILO se encarga de ponerlo en tiempo de arranque.

Las FAQ (Preguntas frecuentemente formuladas) (ver Apéndice A) dan más información sobre LILO, incluyendo como utilizar LILO con el “OS/2’s Boot Manager”.

4.3 Cerrando el Sistema

Cerrar un sistema Linux tiene algo de truco. Recuerde que nunca se debe cortar la corriente o pulsar el botón de reset mientras el sistema esté ejecutándose. El núcleo sigue la pista de la entrada/salida a disco en buffers de memoria. Si se reinicializa el sistema sin darle al núcleo la oportunidad de escribir sus buffers a disco, puede corromper sus sistemas de ficheros.

En tiempo de cierre se toman también otras precauciones. Todos los procesos reciben una señal que les permite morir airosamente (escribiendo y cerrando todos los ficheros y ese tipo de cosas). Los sistemas de ficheros se desmontan por seguridad. Si se desea, el sistema también puede alertar a los usuarios de que se está cerrando y darles la posibilidad de desconectarse.

La forma más simple de cerrar el sistema es con el comando `shutdown`. El formato del comando es

`shutdown` ⟨*tiempo*⟩ ⟨*mensaje-de-aviso*⟩

El argumento ⟨*tiempo*⟩ es el momento de cierre del sistema (en el formato *hh:mm:ss*), y ⟨*mensaje-de-aviso*⟩ es un mensaje mostrado en todos los terminales de usuario antes de cerrar. Alternativamente, se puede especificar el parámetro ⟨*tiempo*⟩ como “`now`”, para cerrar inmediatamente. Se le puede suministrar la opción `-r` a `shutdown` para reinicializar el sistema tras el cierre.

Por ejemplo, para cerrar el sistema a las 8:00pm, se puede utilizar el siguiente comando

```
# shutdown -r 20:00
```

El comando `halt` puede utilizarse para forzar un cierre inmediato, sin ningún mensaje de aviso ni periodo de gracia. `halt` se utiliza si se es el único usuario del sistema y se quiere cerrar el sistema y apagarlo.

◇ No apague o reinicialice el sistema hasta que vea el mensaje:

```
The system is halted
```

Es muy importante que cierre el sistema "limpiamente" utilizando el comando `shutdown` o el `halt`. En algunos sistemas, se reconocerá el pulsar `ctrl-alt-del`, que causará un `shutdown`; en otros sistemas, sin embargo, el utilizar el "Apretón de Cuello de Vulcano" reinicializará el sistema inmediatamente y puede causar un desastre.

4.4 Gestión de Usuarios

Independientemente de que tenga muchos usuarios o no en su sistema, es importante comprender los aspectos de la gestión de usuarios bajo Linux. Incluso si se es el único usuario, se debe tener, presumiblemente, una cuenta distinta de `root` para hacer la mayor parte del trabajo.

Cada persona que utilice el sistema debe tener su propia cuenta. Raramente es una buena idea el que varias personas compartan la misma cuenta. No sólo es un problema de seguridad, sino que las cuentas se utilizan para identificar uní vocamente a los usuarios al sistema. Se necesita ser capaz de saber quién está haciendo qué.

4.4.1 Conceptos de gestión de usuarios

El sistema mantiene una cierta cantidad de información acerca de cada usuario. Dicha información se resume a continuación.

nombre de usuario
El nombre de usuario es el identificador único dado a cada usuario del sistema. Ejemplos de nombres de usuario son `larry`, `karl` y `mdw`. Se pueden utilizar letras y dí gitos junto a los caracteres "_" (subrayado) y "." (punto). Los nombres de usuario se limitan normalmente a 8 caracteres de longitud.

user ID El 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 El group ID, o GID, es la identificación del grupo del usuario por defecto. En la sección 3.9 discutimos los permisos de grupo; cada usuario pertenece a uno o más grupos definidos por el administrador del sistema. Más sobre ésto más adelante.

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
El "nombre real" o "nombre completo" del usuario se almacena junto con el nombre de usuario. Por ejemplo, el usuario `schmoj` puede tener el nombre "Jos Schmo" en la vida real.

directorio inicial
El 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`.

intérprete de inicio
El intérprete de inicio del usuario es el intérprete de comandos que es arrancado para el usuario en tiempo de conexión. Ejemplos pueden ser `/bin/bash` y `/bin/tcsh`.

El fichero `/etc/passwd` contiene la información anterior acerca de los usuarios. Cada lí nea del fichero contiene información acerca de un único usuario; el formato de cada lí nea es

```
nombre:clave encriptada:UID:GID:nombre completo:dir.inicio:intérprete
```

Un ejemplo puede ser:

```
kiwi:Xv8Q981g71oKK:102:100:Laura Poole:/home/kiwi:/bin/bash
```

Como puede verse, el primer campo , "`kiwi`", es el nombre de usuario.

El siguiente campo, "`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 clave secreta. En otras palabras, sólo si se conoce la clave, ésta puede ser desencriptada. Esta forma de encriptación es bastante segura.

Algunos sistemas utilizan "claves en sombra" en la que la información de las claves se relega al fichero `/etc/shadow`. Puesto que `/etc/passwd` es legible por todo el mundo, `/etc/shadow` suministra un grado extra de seguridad, puesto que éste no lo es. Las claves en sombra suministran algunas otras funciones como puede ser la expiración de claves; no entraremos a detallar éstas funciones aquí .

El tercer campo "`102`", es el UID. Este debe ser único para cada usuario. El cuarto campo, "`100`", es el GID. Este usuario pertenece al grupo numerado 100. La información de grupos, como la información de usuarios, se almacena en el fichero `/etc/group`. Véase la sección 4.4.5 para más información.

El quinto campo es el nombre completo del usuario. "`Laura Poole`". Los dos últimos campos son el directorio inicial del usuario (`/home/kiwi`) y el intérprete de conexión (`/bin/bash`), respectivamente. No es necesario que el directorio inicial de un usuario tenga el mismo nombre que el del nombre de usuario. Sin embargo, ayuda a identificar el directorio.

4.4.2 Añadiendo usuarios

Cuando se añade un usuario hay varios pasos a seguir. Primero, se le debe crear una entrada en `/etc/passwd`, con un nombre de usuario y UID únicos. Se debe especificar el GID, nombre completo y resto de información. Se debe crear el directorio inicial, y poner los permisos en el directorio para que el usuario sea el dueño. Se deben suministrar ficheros de comandos de inicialización en el nuevo directorio y se debe hacer alguna otra configuración del sistema (por ejemplo, preparar un buzón para el correo electrónico entrante para el nuevo usuario).

Aunque no es difí cil el añadir usuarios a mano (yo lo hago), cuando se está ejecutando un sistema con muchos usuarios, es fácil el olvidarse de algo. La manera más simple de añadir usuarios es utilizar un programa interactivo que vaya preguntando por la información necesaria y actualice todos los ficheros del sistema automáticamente. El nombre de este programa es `useradd` o `adduser` dependiendo del software que esté instalado. Las páginas `man` para estos comandos deberí an ser suficientemente autoexplicatorias.

4.4.3 Borrando usuarios

De forma parecida, borrar usuarios puede hacerse con los comandos `userdel` o `deluser` dependiendo de qué software fuera instalado en el sistema.

Si se desea "deshabilitar" temporalmente un usuario para que no se conecte al sistema (sin borrar la cuenta del usuario), se puede prefijar con un asterisco ("`*`") el campo de la clave en `/etc/passwd`. Por ejemplo, cambiando la lí nea de `/etc/passwd` correspondiente a `kiwi` a

```
kiwi:*Xv8Q981g71oKK:102:100:Laura Poole:/home/kiwi:/bin/bash
```

evitará que `kiwi` se conecte.

4.4.4 Poniendo atributos de usuario

Después de que haya creado un usuario, puede necesitar cambiar algún atributo de dicho usuario, como puede ser el directorio inicial o la clave. La forma más simple de hacer ésto es cambiar los valores directamente en `/etc/passwd`. Para poner clave a un usuario, utilice el comando `passwd`. Por ejemplo,

```
# passwd larry
```

cambiará la clave de `larry`. Sólo `root` puede cambiar la clave de otro usuario de ésta forma. Los usuarios pueden cambiar su propia clave con `passwd` también.

En algunos sistemas, los comandos `chfn` y `chsh` están disponibles, permitiendo a los usuarios el cambiar sus atributos de nombre completo e intérprete de conexión. Si no, deben pedir al administrador de sistemas que los cambie por ellos.

4.4.5 Grupos

Como hemos citado anteriormente, 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, como dijimos en la sección 3.9, cada fichero tiene un "grupo propietario" y un conjunto de permisos de grupo que define de qué forma pueden acceder al fichero los usuarios del grupo.

Hay varios grupos definidos en el sistema, como pueden ser `bin`, `mail`, y `sys`. Los usuarios no deben pertenecer a ninguno de estos grupos; se utilizan para permisos de ficheros del sistema. En su lugar, los usuarios deben pertenecer a un grupo individual, como `users`. Si se quiere ser detallista, se pueden mantener varios grupos de usuarios como por ejemplo `estudiantes`, `soporte` y `facultad`.

El fichero `/etc/group` contiene información acerca de los grupos. El formato de cada lí nea es

```
nombre de grupo:clave:GID:otros miembros
```

Algunos ejemplos de grupos pueden ser:

```
root:*:0:
usuarios:*:100:mdw,larry
invitados:*:200:
otros:*:250:kiwi
```

El primer grupo, `root`, es un grupo especial del sistema reservado para la cuenta `root`. El siguiente grupo, `users`, es para usuarios normales. Tiene un GID de 100. Los usuarios `mdw` y `larry` tienen acceso a este grupo. Recuérdese que en `/etc/passwd` cada usuario tiene un GID por defecto. Sin embargo, los usuarios pueden pertenecer a más de un grupo, añadiendo sus nombres de usuario a otras lí neas de grupo en `/etc/group`. El comando `groups` lista a qué grupos se tiene acceso.

El tercer grupo, `invitados`, es para usuarios invitados, y `otros` es para "otros" usuarios. El usuario `kiwi` tiene acceso a éste grupo.

Como se puede ver, el campo "clave" de `/etc/group` raramente se utiliza. A veces se utiliza para dar una clave para acceder a un grupo. Esto es raras veces necesario. Para evitar el que los usuarios cambien a grupos privilegiados (con el comando `newgroup`), se pone el campo de la clave a "`*`".

Se pueden usar los comandos `addgroup` o `groupadd` para añadir grupos a su sistema. Normalmente es más sencillo añadir lí neas a `/etc/group` uno mismo, puesto que no se necesitan más configuraciones para añadir un grupo. Para borrar un grupo, sólo hay que borrar su entrada de `/etc/group`.

4.5 Archivando y Comprimiendo Ficheros

Antes de que podamos hablar acerca de copias de seguridad, necesitamos presentar las herramientas utilizadas para archivar ficheros y programas en los sistemas UNIX.

4.5.1 Utilizando `tar`

El comando `tar` es utilizado normalmente para archivar ficheros.

El formato del comando `tar` es

`tar` ⟨*opciones*⟩ ⟨*fichero1*⟩ ⟨*fichero2*⟩ ...⟨*ficheroN*⟩

donde ⟨*opciones*⟩ es la lista de comandos y opciones para `tar`, y ⟨*fichero1*⟩ hasta ⟨*ficheroN*⟩ es la lista de ficheros a añadir o extraer del archivo.

Por ejemplo, el comando

```
# tar cvf backup.tar /etc
```

empaquetará todos los ficheros de `/etc` en el fichero tar `backup.tar`. El primer argumento de `tar` —“`cvf`”— es el “comando” `tar`. “`c`” le dice a `tar` que cree un nuevo fichero de archivo. La opción “`v`” fuerza a `tar` en el modo detallado —imprimiendo los nombres de los ficheros según se archivan. La opción “`f`” le dice a `tar` que el siguiente argumento —`backup.tar`— es el nombre del archivo a crear. El resto de los argumentos de `tar` son los nombres de ficheros y directorios a añadir al archivo.

El comando

```
# tar xvf backup.tar
```

extraerá el fichero tar `backup.tar` en el directorio actual. Esto puede ser peligroso a veces —cuando se extraen ficheros de un fichero tar, los ficheros antiguos se sobreescriben.

Por otra parte, antes de extraer ficheros tar es importante conocer dónde se deben desempaquetar los ficheros. Por ejemplo, digamos que se archivaron los siguientes ficheros: `/etc/hosts`, `/etc/group`, y `/etc/passwd`. Si se usó el comando

```
# tar cvf backup.tar /etc/hosts /etc/group /etc/passwd
```

el nombre de directorio `/etc` se añadió al principio de cada nombre de fichero. Para poder extraer los ficheros en la localización correcta, se necesitará utilizar los siguientes comandos:

```
# cd /
# tar xvf backup.tar
```

Puesto que los ficheros se extraen con el nombre de camino almacenado en el fichero de archivo.

Sin embargo, si se archivaron los ficheros con los comandos

```
# cd /etc
# tar cvf hosts group passwd
```

Los nombres de directorio no se salvaron en el fichero de archivo. Por ésto se necesitará hacer "`cd /etc`" antes de extraer los ficheros. Como se puede ver, el cómo haya sido creado un fichero tar marca una gran diferencia en cómo se extrae. Se puede usar el comando

```
# tar tvf backup.tar
```

para mostrar un "índice" del fichero tar antes de desempaquetarlo. De esta forma se puede ver que directorio se utilizó como origen de los nombres de los ficheros, y se puede extraer el archivo desde la localización correcta.

4.5.2 gzip y compress

A diferencia de los programas de archivo para MS-DOS, `tar` no comprime automáticamente los ficheros según los archiva. Por ello, si se están archivando dos ficheros de un megabyte, el fichero tar resultante tendrá dos megabytes. El comando `gzip` puede utilizarse para comprimir un fichero (el fichero a comprimir no necesita ser un fichero tar). El comando

```
# gzip -9 backup.tar
```

comprimirá `backup.tar` y le dejará con `backup.tar.gz`, que es la versión comprimida del fichero. La opción `-9` le dice a `gzip` que utilice el mayor factor de compresión.

El comando `gunzip` puede ser utilizado para descomprimir un fichero comprimido con `gzip`. Equivaléntemente, se puede utilizar "`gzip -d`".

`gzip` es una herramienta relativamente nueva en la comunidad UNIX. Durante muchos años, se utilizó en su lugar el comando `compress`. Sin embargo, debido a varios factores[1], `compress` se está volviendo desfasado.

Los ficheros comprimidos con `compress` terminan en la extensión `.Z`. Por ejemplo, `backup.tar.Z` es la versión comprimida con `compress` de `backup.tar`, mientras que `backup.tar.gz` es la versión comprimida con `gzip`[2]. El comando `uncompress` se utiliza para expandir un fichero comprimido con `compress`; `gunzip` sabe también como tratar los ficheros comprimidos con `compress`.

4.5.3 Juntándolo todo

Por lo tanto, para archivar un grupo de ficheros y comprimir el resultado, se pueden utilizar los comandos:

```
# tar cvf backup.tar /etc
# gzip -9 backup.tar
```

[1] Estos factores incluyen una disputa por una patente software contra el algoritmo de `compress` y el hecho de que `gzip` es mucho más eficiente que `compress`.

[2] Para añadir más confusión, durante algún tiempo la extensión `.z` ("z" minúscula) fué utilizada para los ficheros comprimidos con `gzip`. La extensión oficial de los ficheros `gzip` ahora es `.gz`.

El resultado será `backup.tar.gz`. Para desempaquetar este fichero, se usan los comandos contrarios:

```
# gunzip backup.tar.gz
# tar xvf backup.tar
```

Por supuesto, asegúrese siempre de que está en el directorio correcto antes de desempaquetar un fichero tar.

Se pueden utilizar algunas mañas UNIX para hacer todo esto en una sola línea de comando, como por ejemplo:

```
# tar cvf - /etc | gzip -9c > backup.tar.gz
```

Aquí estamos enviando el fichero tar a "`-`", que representa la salida estándar de `tar`. Esto es encolado hacia `gzip`, que comprime el fichero tar de entrada, y el resultado se salva en `backup.tar.gz`. La opción `-c` de `gzip` le dice que envíe su salida hacia la salida estándar, que es redirigida a `backup.tar.gz`.

Un comando único para desempaquetar este archivo sería:

```
# gunzip -c backup.tar.gz | tar xvf -
```

De nuevo, `gunzip` descomprime el contenido de `backup.tar.gz` y envía el fichero tar resultante hacia la salida estándar. Esta es redirigida hacia `tar`, que lee de "`-`", refiriéndose ésta vez a la entrada estándar de `tar`.

Felizmente, el comando `tar` también incluye la opción `z` para comprimir/descomprimir automáticamente los ficheros al vuelo, utilizando el algoritmo de compresión de `gzip`.

Por ejemplo, el comando

```
# tar cvfz backup.tar.gz /etc
```

es equivalente a

```
# tar cvf backup.tar /etc
# gzip backup.tar
```

Igual que el comando

```
# tar xvfz backup.tar.Z
```

se puede utilizar en vez de

```
# uncompress backup.tar.Z
# tar xvf backup.tar
```

Remítase a las páginas man para obtener mas información acerca de `tar` y `gzip`.

4.6 Usando Disquetes y Haciendo Copias de Seguridad

Los disquetes son utilizados normalmente como medio para copias de seguridad. Si no se tiene una unidad de cinta conectada al sistema, se pueden utilizar disquetes (a pesar de que sean más lentos y ligéramente menos seguros).

También puede utilizar disquetes para contener sistemas de ficheros individuales —de ésta forma, se puede montar **mount** el disquete para acceder a los datos contenidos en él.

4.6.1 Utilizando disquetes para copias de seguridad

La forma más simple de hacer una copia de seguridad es con `tar`. El comando

```
# tar cvfzM /dev/fd0 /
```

hará una copia de seguridad completa de su sistema utilizando el disquete `/dev/fd0`. La opción "M" de `tar` permite que la copia de seguridad sea una copia multi-volumen; esto es, cuando un disquete está lleno, `tar` pedirá el siguiente. El comando

```
# tar xvfzM /dev/fd0
```

puede ser utilizado para recuperar la copia de seguridad completa. Este método puede ser utilizado también si se tiene una unidad de cinta (`/dev/rmt0`) conectada al sistema.

Existen otros programas para hacer copias de seguridad multi-volumen; el programa `backflops` disponible en `tsx-11.mit.edu` puede ser útil.

Hacer una copia de seguridad completa del sistema puede ser costoso en tiempo y recursos. Muchos administradores de sistemas utilizan una política de copias de seguridad incrementales, en la que cada mes se hace una copia de seguridad completa, y cada semana sólo se copian aquellos ficheros que hayan sido modificados en esa semana. En este caso, si el sistema se viene abajo a mitad de mes, sólo tiene que restaurar la última copia de seguridad mensual completa y, después, las últimas copias semanales según el caso.

El comando `find` puede ser útil para localizar ficheros que hayan cambiado desde una cierta fecha. Se pueden encontrar varios ficheros de comandos para manejar copias de seguridad incrementales en `sunsite.unc.edu`.

4.6.2 Utilizando disquetes como sistemas de ficheros

Puede crearse un sistema de ficheros en un disquete igual que lo haría en una partición de un disco duro. Por ejemplo,

```
# mke2fs /dev/fd0 1440
```

crea un sistema de ficheros en el disquete en /dev/fd0. El tamaño del sistema de ficheros debe corresponder al tamaño del disquete. Los disquetes de alta densidad de 3.5" tienen un tamaño de 1.44 megabytes, o 1440 bloques. Los disquetes de alta densidad de 5.25" tienen 1200 bloques.

Para poder acceder a un disquete, se debe montar **mount** el sistema de ficheros que contiene. El comando

```
# mount -t ext2 /dev/fd0 /mnt
```

montará el disquete en /dev/fd0 en el directorio /mnt. Ahora todos los ficheros del disquete aparecerán bajo /mnt en su unidad. "-t ext2" especifica el tipo de sistema de ficheros como ext2fs. Si crea otro tipo de sistema de ficheros en el disquete, necesitará especificárselo al comando mount.

El "punto de montaje" (el directorio donde está montando el sistema de ficheros) debe existir en el momento de utilizar el comando mount. Si no existiese, deba crearlo con el comando mkdir.

Para más información sobre sistemas de ficheros, montaje y puntos de montaje, ver sección 4.8.

◇ Tenga en cuenta que cualquier entrada/salida al disquete se gestiona con buffers igual que si fuese de disco duro. Si cambia datos en el disquete, puede que no vea encenderse la luz de la unidad hasta que el núcleo decida vaciar sus buffers. Es importante que no quite un disquete antes de haberlo desmontado; esto puede hacerse con el comando

```
# umount /dev/fd0
```

No cambie los disquetes como se hace en un sistema MS-DOS; siempre que cambie disquetes, desmonte umount el primero y monte mount el siguiente.

4.7 Actualizando e Instalando Nuevo Software

Otra tarea del administrador de sistemas es la actualización e instalación de nuevo software.

La comunidad Linux es muy dinámica. Las versiones nuevas del núcleo aparecen cada pocas semanas, y otros programas se actualizan casi tan a menudo. Por esto, los nuevos usuarios de Linux sienten a menudo la necesidad de actualizar sus sistemas constantemente para mantener el paso de los cambios. No sólo ésto no es necesario, sino que es una pérdida de tiempo. Para mantenerse al dí a de todos los cambios del mundo Linux, debería utilizar todo su tiempo actualizando en vez de usando su sistema.

Pero entonces, ¿cuándo se debe actualizar?. Alguna gente piensa que se debe actualizar cuando se libera una nueva versión de distribución —por ejemplo, cuando Slackware presenta una nueva versión. Muchos usuarios Linux reinstalan completamente sus sistemas con la nueva versión Slackware, cada vez. Esto, también, es una pérdida de tiempo. En general, los cambios de las versiones Slackware son pequeños. Bajarse y reinstalar 30 discos cuando sólo ha cambiado un 10% del software es, por supuesto, inútil.

La mejor forma de actualizar su sistema es haciéndolo a mano: actualizando solo aquellos paquetes de software que sepa que hay que actualizar. Esto asusta a un montón de gente: quieren saber qué actualizar y cómo, y qué se romperá si no actualizan. Para tener éxito con Linux, es importante superar los temores del "hágalo usted mismo" —que es lo que Linux es, a fin de cuentas. De hecho, una vez que tenga su sistema trabajando y todo el software correctamente configurado, la reinstalación de una nueva versión no dudará en limpiar toda la configuración y todo estará roto otra vez, igual que la primera vez que instaló su sistema. Volver atrás de ésta forma no es necesario —todo lo que se necesita es algún conocimiento acerca de cómo actualizar el sistema y de cómo hacerlo bien.

Se encontrará con que cuando actualice un componente de su sistema, no tienen por que fallar los demás. Por ejemplo, la mayor parte del software de mi sistema se quedó en una instalación antigua basada en el nivel 0.96 MCC Interim. Ciertamente, utilizo la versión más reciente del núcleo y de las librerías con éste software sin problemas. Realmente, la actualización insensata para "mantenerse en la ola" no es tan importante. Esto no es MS-DOS o Microsoft Windows. No hay ninguna razón importante para ejecutar la versión más reciente de todo el software. Si encuentra que desea o necesita funciones de una versión nueva, entonces actualice. Si no, no lo haga. En otras palabras, actualice sólo lo que tenga que hacerlo y cuando tenga que hacerlo. No actualice sólo por el mero hecho de actualizar. Hacerlo sólo gastaría un montón de tiempo y esfuerzo intentando mantenerse al día.

El software más importante para actualizar en su sistema es el núcleo, las librerías y el compilador `gcc`. Estas son las tres partes esenciales de su sistema, y en algunos casos cada uno depende de las otras para que todo funcione bien. La mayor parte del resto del software de su sistema no necesita ser actualizado periódicamente.

4.7.1 Actualizando el núcleo

Actualizar el núcleo es simplemente asunto de coger los fuentes y compilarlos uno mismo. Debe compilar el núcleo por si mismo para poder activar o desactivar ciertas funciones, además de asegurarse de que el núcleo estará optimizado para correr en su máquina. El proceso es casi indoloro.

Los fuentes del núcleo pueden obtenerse de cualquiera de los servidores FTP de Linux (para una lista, véase Apéndice C). En `sunsite.unc.edu`, por ejemplo, los fuentes del núcleo se encuentran en `/pub/Linux/kernel`. Las versiones del núcleo se numeran utilizando un número de versión y un nivel de "parche". Por ejemplo, la versión de núcleo 0.99, nivel de parche 11 es nombrada como `0.99.pl11`, o simplemente `0.99.11`.

Los fuentes del núcleo se presentan en un fichero tar comprimido con `gzip`[3]. Por ejemplo, el fichero conteniendo los fuentes del núcleo 0.99.pl11 es `linux-0.99.11.tar.gz`.

Se desempaqueta este fichero tar desde el directorio `/usr/src`; esto crea el directorio `/usr/src/linux` que contiene los fuentes del núcleo. Se debe borrar o cambiar de nombre el directorio `/usr/src/linux` existente antes de desempaquetar la nueva versión.

Una vez que se han desempaquetado los fuentes, es necesario asegurarse que dos enlaces simbólicos son correctos. Para crear estos enlaces, se utilizan los comandos:

[3] A menudo, se suministra un fichero de parches para la versión actual del núcleo, que permite parchear los fuentes actuales desde el nivel anterior al actual (utilizando el programa `patch`). En muchos casos, sin embargo, es normalmente más sencillo el instalar la nueva versión de los fuentes del núcleo completa.

```
# ln -sf /usr/src/linux/include/linux /usr/include/linux
# ln -sf /usr/src/linux/include/asm /usr/include/asm
```

Una vez creados estos enlaces una vez, no hay razón para crearlos de nuevo cuando se instale la siguiente versión de los fuentes del núcleo. (Véase sección 3.10 para más información sobre enlaces simbólicos.)

Hay que advertir, que para poder compilar el núcleo, se debe tener `gcc` y `g++`, compiladores C y C++, instalados en el sistema. Puede necesitar tener la versión más reciente de estos compiladores: ver sección 4.7.3, más adelante, para más información.

Para compilar el núcleo, primero cambie de directorio usando `cd` a `/usr/src/linux`. Ejecute el comando `make config`. Este comando le irá preguntando por un número de opciones de configuración, como puede ser qué tipos de sistemas de ficheros se desea incluir en el nuevo núcleo.

Después, edite `/usr/src/linux/Makefile`. Asegúrese que la definición para `ROOT_DEV` sea correcta —define el dispositivo utilizado como sistema de ficheros raíz en tiempo de arranque. La definición normal es

```
ROOT_DEV = CURRENT
```

A menos que esté cambiando el dispositivo raíz del sistema de ficheros, no hay razón para cambiar esto.

Después, ejecute el comando `make dep` para asegurar que se cumplen todas las dependencias del fuente. Este es un paso muy importante.

Finalmente, se está listo para compilar el núcleo. El comando `make Image` compilará el núcleo y dejará la nueva imagen del núcleo en el fichero `/usr/src/linux/Image`. Alternativamente, el comando `make zImage` compilará una imagen comprimida del núcleo, que se descomprime a sí misma en tiempo de arranque y utiliza menos espacio en el disco.

Una vez que está el núcleo compilado, se necesita, o bien copiarlo a un disquete de arranque (con un comando como "`cp Image /dev/fd0`") o bien instalarlo utilizando LILO para arrancar desde el disco duro. Ver sección 4.2.2 para más información.

4.7.2 Actualizando las librerías

Como se mencionó antes, la mayor parte del software del sistema está compilado para que utilice las librerías compartidas, que contienen subrutinas comunes compartidas entre distintos programas.

Si le aparece el mensaje

```
Incompatible library version
```

cuando se intenta ejecutar un programa, entonces necesita actualizar a la versión de las librerías que el programa requiere. Las librerías son compatible-ascendentes; esto es, un programa compilado para utilizar una versión antigua de las librerías, debe trabajar con la nueva versión de las librerías instalada. Sin embargo, lo contrario no es cierto.

La última versión de las librerías se puede encontrar en los servidores FTP de Linux. En `sunsite.unc.edu`, están disponibles en `/pub/Linux/GCC`. Los ficheros de "versión" que se encuentran ahí deben explicar qué ficheros se necesita obtener y cómo instalarlos. Rápidamente, se deben coger los ficheros `image-`*versión*`.tar.gz` y `inc-`*versión*`.tar.gz` donde *versión* es la versión de las librerías a instalar, por ejemplo `4.4.1`. Son ficheros tar comprimidos con `gzip`; el fichero `image` contiene las imágenes de las librerías a instalar en `/lib` y `/usr/lib`. El fichero `inc` contiene los ficheros de inclusión a instalar en `/usr/include`.

El fichero `release-`*versión*`.tar.gz` debe explicar el método de instalación detalladamente (las instrucciones exactas varían para cada versión). En general, se necesita instalar los ficheros de librerías `.a` y `.sa` en `/usr/lib`. Estas son las librerías utilizadas en tiempo de compilación.

Además, los ficheros imagen de las librerías compartidas `libc.so.`*versión* se instalan en `/lib`. Estas son las imágenes de las librerías compartidas que son cargadas en tiempo de ejecución por los programas que utilizan las librerías. Cada librería tiene un enlace simbólico utilizando el número de versión mayor de la librería en `/lib`

Por ejemplo, la versión 4.4.1 de la librería `libc` tiene un número de versión mayor de `4`. El fichero que contiene la librería es `libc.so.4.4.1`. Existe un enlace simbólico del nombre `libc.so.4` en `/lib` apuntando a este fichero. Por ejemplo, cuando se actualiza de `libc.so.4.4` a `libc.so.4.4.1`, necesita cambiar el enlace simbólico para apuntar a la nueva versión.

◇ Es muy importante que se cambie el enlace simbólico en un solo paso, como se indica más abajo. Si de alguna forma borrase el enlace simbólico `libc.so.4`, los programas que dependen del enlace (incluyendo utilidades básicas como `ls` y `cat`) dejarán de funcionar. Utilice el siguiente comando para actualizar el enlace simbólico `libc.so.4` para que apunte al fichero `libc.so.4.4.1`:

```
# ln -sf /lib/libc.so.4.4.1 /lib/libc.so.4
```

Se necesita también cambiar el enlace simbólico `libm.so.`*versión* de la misma forma. Si se está actualizando a una versión diferente de las librerías, sustituya lo anterior con los nombres adecuados. La nota de la versión de la librería debe explicar los detalles. (Ver sección 3.10 para más información acerca de los enlaces simbólicos).

4.7.3 Actualizando gcc

El compilador de C y C++ `gcc` se utiliza para compilar software en su sistema, siendo el más importante el núcleo. La versión más reciente de `gcc` se encuentra en los servidores FTP de Linux. En `sunsite.unc.edu` se encuentra en el directorio `/pub/Linux/GCC` (junto con las librerías). Debe existir un fichero `release` para la distribución `gcc` detallando qué ficheros se necesitan obtener y como instalarlos.

4.7.4 Actualizando otro software

La actualización de otro software sólo suele consistir en obtener los ficheros apropiados e instalarlos. Mucho software para Linux se distribuye como ficheros tar comprimidos con `gzip`, incluyendo, ya sea fuentes, ejecutables o ambos. Si los ejecutables no se incluyen con la versión, puede ser necesario recompilarlos uno mismo; normalmente esto significa teclear `make` en el directorio que tiene los fuentes.

Leer el newsgroup de USENET `comp.os.linux.announce` en busca de anuncios de nuevas versiones de software es la mejor manera de enterarse de la aparición de nuevo software. Siempre que busque software en un servidor FTP bájese el fichero de índice `ls-lR` del servidor FTP y utilice `grep` para encontrar los ficheros en cuestión, es la forma más simple de localizar software. Si tiene la posibilidad de utilizar `archie`, éste puede servirle de ayuda. Si no tiene `archie`, puede conectarse con `telnet` a un servidor `archie` como puede ser `archie.rutgers.edu`, identificarse como "`archie`" y utilizar el comando "`help`". Véase Apéndice A para más detalles.

Una fuente útil de software para Linux, son las imágenes de los discos de la distribución Slackware. Cada disco contiene varios ficheros `.tgz` que son simples ficheros tar comprimidos con `gzip`. En vez de bajarse los discos, puede bajar sólo los ficheros `.tgz` deseados de los directorios Slackware del servidor FTP e instalarlos directamente. Si puede ejecutar la distribución Slackware, puede utilizar el comando `setup` para cargar e instalar automáticamente una serie completa de discos.

Insistimos, normalmente no es una buena idea el actualizar reinstalando desde cero la versión más reciente de Slackware, u otra distribución. Si se reinstala de esta forma, no lo dude, arruinará su instalación actual. La mejor forma de actualizar software es por partes; esto es, si hay un programa que se usa a menudo y tiene una nueva versión, actualícelo. Si no, no se moleste. Regla práctica: Si no tiene nada roto, no lo repare. Si su software actual funciona, no hay motivo para actualizar.

4.8 Gestionando Sistemas de Ficheros

Otra tarea del administrador de sistemas es el cuidado de los sistemas de ficheros. Mucho de éste trabajo se basa en comprobaciones periódicas del sistema de ficheros en busca de ficheros dañados o corrompidos; muchos sistemas comprueban automáticamente los sistemas de ficheros en tiempo de arranque.

4.8.1 Montando sistemas de ficheros

Para empezar, algunos conceptos acerca de sistemas de ficheros. Antes de que un sistema de ficheros sea accesible al sistema, debe ser **montado** en algún directorio. Por ejemplo, si se tiene un sistema de ficheros en un disquete, se debe montar bajo algún directorio, digamos `/mnt`, para poder acceder a los ficheros que contiene (véase la sección 4.6.2). Tras montar el sistema de ficheros, todos los ficheros en dicho sistema aparecen en ese directorio. Tras desmontar el sistema de ficheros, el directorio (en este caso, `/mnt`) estará vacío.

Lo mismo es válido para los sistemas de ficheros del disco duro. El sistema monta automáticamente los sistemas de ficheros del disco duro en tiempo de arranque. El así llamado "sistema de ficheros raíz" es montado en el directorio `/`. Si se tiene un sistema de ficheros separado para `/usr`, por ejemplo, se monta en `/usr`. Si sólo se tiene un sistema de ficheros raíz, todos los ficheros (incluyendo los de `/usr`) existen en ese sistema de ficheros.

El comando `mount` se utiliza para montar un sistema de ficheros. El comando

```
mount -av
```

se ejecuta desde el fichero `/etc/rc` (que es el fichero de inicialización del sistema, ejecutado en tiempo de arranque; véase la sección 4.10.1). El comando `mount -av` obtiene información de los sistemas de ficheros y puntos de montaje del fichero `/etc/fstab`. Este es un ejemplo de fichero `fstab`:

```
# dispositivo  directorio    tipo    opciones
/dev/hda2       /            ext2    defaults
/dev/hda3       /usr         ext2    defaults
/dev/hda4       none         swap    sw
/proc           /proc        proc    none
```

El primer campo es el dispositivo —el nombre de la partición a montar. El segundo campo es el punto de montaje. El tercero es el tipo de sistema de ficheros —como puede ser `ext2` (para ext2fs) o `minix` (para sistemas de ficheros Minix). La tabla 4.1 lista los distintos tipos de sistemas de ficheros disponibles en Linux.[4] Puede que no todos éstos tipos de sistemas de ficheros estén disponibles en su sistema; el núcleo debe tener soporte para ellos compilado en él. Véase sección 4.7 para información sobre cómo construir un núcleo.

Sistema de ficheros	Nombre de tipo	Comentarios
Second Extended Filesystem	`ext2`	Sistema de ficheros mas común en Linux.
Extended Filesystem	`ext`	Reemplazado por `ext2`.
Minix Filesystem	`minix`	Sistema de ficheros Minix original; raras veces utilizado.
Xia Filesystem	`xia`	Como `ext2`, pero raras veces utilizado.
UMSDOS Filesystem	`umsdos`	Utilizado para instalar Linux en una partición MS-DOS.
MS-DOS Filesystem	`msdos`	Utilizado para acceder a ficheros MS-DOS.
`/proc` Filesystem	`proc`	Suministra información de proceso para `ps`, etc.
ISO 9660 Filesystem	`iso9660`	Formato utilizado por muchos CD-ROMs.
Xenix Filesystem	`xenix`	Sistema de ficheros de Xenix.
System V Filesystem	`sysv`	Variantes del System V para el x86.
Coherent Filesystem	`coherent`	Acceso a ficheros de Coherent.
HPFS Filesystem	`hpfs`	Acceso en lectura a particiones HPFS (DoubleSpace).

Tabla 4.1: Tipos de Sistemas de Ficheros en Linux

El último campo del fichero `fstab` contiene las opciones del comando `mount` —normalmente, está puesto a "`defaults`" (defecto).

Como se puede ver, las particiones de intercambio están incluidas en `/etc/fstab` también. Tienen un punto de montaje de `none` y tipo `swap`. El comando `swapon -a`, que se ejecuta también desde `/etc/rc`, se utiliza para activar el intercambio en todos los dispositivos de intercambio listados en `/etc/fstab`.

[4] Esta tabla es válida a la versión de núcleo 1.1.37.

El fichero `fstab` contiene una entrada especial —para el sistema de ficheros `/proc`. Tal y como se mencionó en la sección 3.11.1, el sistema de ficheros `/proc` se utiliza para almacenar información acerca de los procesos del sistema, memoria disponible, y otros datos del mismo tipo. Si `/proc` no está montado, no funcionarán comandos como `ps`.

◇ El comando `mount` sólo puede ser utilizado por root. Esto es así para garantizar la seguridad del sistema; no es deseable que usuarios normales estén montando y desmontando sistemas de ficheros a su antojo. Existen varios paquetes disponibles que permiten a los usuarios normales montar y desmontar sistemas de ficheros (disquetes en particular) sin comprometer la seguridad del sistema.

El comando `mount -av` realmente monta todos los sistemas de ficheros excepto el sistema de ficheros raíz (en la tabla anterior, `/dev/hda2`). El sistema de ficheros raíz es montado automáticamente en tiempo de arranque por el núcleo.

En vez de utilizar el comando `mount -av`, se puede montar un sistema de ficheros a mano. El comando

`#` *mount -t ext2 /dev/hda3 /usr*

es equivalente a montar el sistema de ficheros con la entrada `/dev/hda3` del ejemplo de fichero `fstab` anterior.

En general, nunca se debe montar o desmontar sistemas de ficheros a mano. El comando `mount -av` en `/etc/rc` se encarga de montar los sistemas de ficheros en tiempo de arranque. Los sistemas de ficheros son desmontados por los comandos `shutdown` o `halt` antes de cerrar el sistema.

4.8.2 Comprobando sistemas de ficheros

Normalmente es una buena idea el comprobar de vez en cuando los sistemas de ficheros en busca de ficheros dañados o corrompidos. Algunos sistemas comprueban automáticamente sus sistemas de ficheros en tiempo de arranque (con los comandos apropiados en `/etc/rc`).

El comando utilizado para comprobar un sistema de ficheros depende del tipo de sistema de ficheros en cuestión. Para sistemas de ficheros ext2fs (el tipo más utilizado normalmente), el comando es `e2fsck`. Por ejemplo, el comando

`#` *e2fsck -av /dev/hda2*

comprobará el sistema de ficheros ext2fs de `/dev/hda2` y corregirá automáticamente cualquier error.

Normalmente es una buena idea el desmontar un sistema de ficheros antes de comprobarlo. Por ejemplo, el comando

`#` *umount /dev/hda2*

desmontará el sistema de ficheros en `/dev/hda2`, tras lo cual podrá ser comprobado. La única excepción es que no se puede desmontar el sistema de ficheros raíz. Para poder comprobar el sistema de ficheros raíz cuando está desmontado, se debe utilizar un disquete de arranque/raíz (véase la sección 4.11.1). Tampoco se puede desmontar un sistema de ficheros si alguno de sus ficheros está "ocupado" —esto es, siendo utilizado por un proceso en ejecución. Por ejemplo, no se puede desmontar un sistema de ficheros si el directorio de trabajo de algún usuario está en ese sistema de ficheros. Se recibirá un error "`Device busy`" (dispositivo ocupado) si se intenta desmontar un sistema de ficheros que esté en uso.

Otros tipos de sistemas de ficheros utilizan formas diferentes del comando `e2fsck`, como pueda ser `efsck` y `xfsck`. En algunos sistemas, se puede utilizar el comando `fsck`, que determina el tipo de sistema de ficheros y ejecuta el comando apropiado.

◇ Es importante que se reinicialice el sistema inmediatamente después de comprobar un sistema de ficheros montado, si es que se hizo alguna corrección al sistema de ficheros. (Sin embargo, en general, no se deben comprobar sistemas de ficheros que estén montados.) Por ejemplo, si `e2fsck` informa que ha corregido algún error en el sistema de ficheros, se debe apagar el sistema con `shutdown -r` para rearrancarlo. Esto permite al sistema resincronizar su información acerca del sistema de ficheros cuando `e2fsck` lo modifica.

El sistema de ficheros `/proc` no necesita nunca ser comprobado de esta forma. `/proc` es un sistema de ficheros en memoria, gestionado directamente por el núcleo.

4.9 Utilizando un fichero de intercambio

En vez de reservar una partición individual para espacio de intercambio, se puede usar un fichero. Sin embargo, hay que instalar todo Linux y hacer todas las tareas de configuración *antes* de crear el fichero de intercambio.

Si tiene un sistema Linux instalado, puede utilizar los siguientes comandos para crear un fichero de intercambio. A continuación vamos a crear un fichero de intecambio de 8208 bloques de tamaño (unos 8 megas).

```
# dd if=/dev/zero of=/swap bs=1024 count=8208
```

Este comando crea el fichero de intercambio propiamente dicho. Reemplace "`count=`" con el tamaño del fichero de intercambio en bloques.

```
# mkswap /swap 8208
```

Este comando inicializará el fichero de intercambio `swap`; como antes, se debe cambiar el nombre y tamaño del fichero de intercambio a los valores apropiados.

```
# /etc/sync
# swapon /swap
```

Ahora, tras sincronizar, lo que asegura que el fichero haya sido escrito a disco, ya estamos utilizando el fichero `/swap` que habíamos creado como fichero de intercambio.

El principal inconveniente de utilizar un fichero de intercambio de ésta forma es que todos los accesos al fichero de intercambio se realizan a través del sistema de ficheros. Esto significa que los bloques que forman el fichero de intercambio pueden no ser contiguos. Por esto, el rendimiento no puede ser tan alto como cuando se utiliza una partición de intercambio, en la que los bloques están siempre contiguos y las peticiones de entrada/salida se hacen directamente al dispositivo.

Otra desventaja a la hora de utilizar un fichero de intercambio es la posibilidad de corromper los datos del sistema de ficheros —cuando se utilizan grandes ficheros de intercambio, existe la posibilidad de que se pueda corromper el sistema de ficheros si algo va mal. Manteniendo los sistemas de ficheros y particiones de intercambio separados prevendrá la posibilidad de que esto llegue a ocurrir.

La utilización de un fichero de intercambio puede ser muy útil si se tiene una necesidad temporal de más espacio de intercambio. Por ejemplo, si se está compilando un gran programa y se quiere acelerar las cosas, se puede crear un fichero de intercambio de forma temporal y utilizarlo conjuntamente con el espacio habitual de intercambio.

Para eliminar un fichero de intercambio, primero se usa `swapoff` de esta forma

```
# swapoff /swap
```

Y ahora se puede borrar de forma segura el fichero.

```
# rm /swap
```

Recuérde que cada fichero de intercambio (o partición) puede ser de hasta 16 megabytes, pero se pueden usar hasta 8 ficheros o particiones de intercambio en un sistema.

4.10 Tareas Varias

Lo crea o no, hay un número de tareas domésticas para el administrador de sistemas que no caen en ninguna categoría principal.

4.10.1 Ficheros de arranque del sistema

Cuando el sistema arranca, se ejecutan automáticamente una serie de ficheros de comandos (scripts) en el sistema, antes de que ningún usuario entre. Aquí tenemos una descripción de lo que ocurre:

En tiempo de arranque, el núcleo arranca el proceso `/etc/init`. `init` es un programa que lee su fichero de configuración, `/etc/inittab`, y arranca otros procesos basados en el contenido de este fichero. Uno de los procesos más importantes arrancado desde `inittab` es el proceso `/etc/getty`, arrancado en cada consola virtual. El proceso `getty` dispone la consola virtual para ser utilizada y arranca un proceso `login` en ella. Esto le permite conectarse a cada consola virtual; si `/etc/inittab` no contiene un proceso `getty` para una consola virtual determinada, no se podrá conectar nadie a ella.

Otro proceso ejecutado desde `/etc/inittab` es `/etc/rc`, el fichero de inicialización principal del sistema. Este fichero es simplemente un fichero de comandos que ejecuta cualquier comando de inicialización necesario en tiempo de arranque, como es montar los sistemas de ficheros (véase sección 4.8) e inicializar el espacio de intercambio (memoria virtual).

Su sistema puede ejecutar otros ficheros de comandos de inicialización también, como puede ser `/etc/rc.local`. `/etc/rc.local` contiene normalmente comandos de inicialización específicos de su sistema, como puede ser el establecimiento del nombre del ordenador (véase la siguiente sección). `rc.local` puede ser arrancado desde `/etc/rc` o directamente desde `/etc/inittab`.

4.10.2 Estableciendo el nombre del ordenador

En un entorno de red el nombre del ordenador es utilizado para identificar unívocamente una máquina particular, mientras que en un entorno autónomo, el nombre del ordenador da a la máquina personalidad y encanto. Es como darle nombre a un animal doméstico: siempre puede dirigirse a su perro como "El perro", pero es mucho más interesante asignarle al perro un nombre como "Mancha" o "Tes". Poner el nombre del sistema se limita a utilizar el comando `hostname`. Si se está en una red, su nombre debe ser el nombre completo de su máquina, por ejemplo, `goober.norelco.com`. Si no se está en una red de ningún tipo, se pueden escoger nombre de ordenador y de dominio arbitrarios, como por ejemplo `loomer.vpizza.com`, `shoop.nowhere.edu`, o `floof.org`.

Cuando se pone el nombre del ordenador, dicho nombre debe aparecer en el fichero `/etc/hosts`, que asigna una dirección IP a cada ordenador. Incluso si su ordenador no está en una red, se debe incluir el nombre del ordenador en `/etc/hosts`.

Por ejemplo, si no se está en una red TCP/IP, y el nombre del ordenador es `floof.org`, incluya la línea siguiente en `/etc/hosts`:

```
127.0.0.1        floof.org localhost
```

Esto asigna el nombre del ordenador, `floof.org`, a la dirección de bucle 127.0.0.1 (utilizada si no se está en una red). El alias `localhost` se asigna también a dicha dirección.

Si se está en una red TCP/IP, sin embargo, su dirección y nombre de ordenador real deben aparecer en `/etc/hosts`. Por ejemplo, si su nombre de ordenador es `goober.norelco.com` y su dirección IP es 128.253.154.32, añada la siguiente línea a `/etc/hosts`:

```
128.253.154.32        goober.norelco.com
```

Si el nombre de su ordenador no aparece en `/etc/hosts` no será posible establecerlo.

Para establecer el nombre de su ordenador, utilice el comando `hostname`. Por ejemplo, el comando

`#` *hostname -S goober.norelco.com*

pone el nombre del ordenador a `goober.norelco.com`. En muchos casos, el comando `hostname` se ejecuta en alguno de los ficheros de inicialización del sistema, como puede ser `/etc/rc` ó `/etc/rc.local`. Edite estos ficheros y cambie el comando `hostname` existente para poner su propio nombre de ordenador; al rearrancar el sistema, el nombre del ordenador cambiará al nuevo valor.

4.11 Qué Hacer En Una Emergencia

En algunas ocasiones, el administrador de sistemas se encuentra con el problema de recuperarse de un desastre completo, como puede ser el olvidarse la palabra clave del usuario raíz, o el enfrentarse con sistemas de ficheros dañados. El mejor consejo es, *obrar sin pánico.* Todo el mundo comete errores estúpidos —ésta es la mejor forma de aprender sobre administración de sistemas: la forma difícil.

Linux no es una versión inestable de UNIX. De hecho, he tenido menos problemas con cuelgues de sistemas Linux que con versiones comerciales de UNIX en muchas plataformas. Linux también se beneficia de un fuerte complemento de asistentes que pueden ayudar a salir del agujero.

El primer paso al investigar cualquier problema es intentar arreglarlo uno mismo. Hurgue alrededor, vea cómo funcionan las cosas. Demasiadas veces, un administrador de sistemas pondrá un mensaje desesperado rogando ayuda antes de investigar el problema. Muchas de las veces, encontrará que arreglar problemas por uno mismo es realmente muy fácil. Este es el camino que debe seguir para convertirse en un gurú.

Hay pocos casos en los que sea necesario reinstalar el sistema desde cero. Muchos nuevos usuarios borran accidentalmente algún fichero esencial del sistema, e inmediatamente acuden a los discos de instalación. Esta no es una buena idea. Antes de tomar medidas drásticas como esa, investigar el problema y preguntar a otros ayudará a solucionar las cosas. En prácticamente todos los casos, podrá recuperar el sistema desde un disquete de mantenimiento.

4.11.1 Recuperación utilizando un disquete de mantenimiento

Una herramienta indispensable para el administrador de sistemas es el llamado "disco arranque/raíz" ("boot/root disk") —un disquete desde el que se puede arrancar un sistema Linux completo, independiente del disco duro. Los discos de arranque/raíz son realmente muy simples —se crea un sistema de ficheros raíz en el disquete, se ponen todas las utilidades necesarias en él y se instala LILO y un núcleo arrancable en el disquete. Otra técnica es usar un disquete para el núcleo y otro para el sistema de ficheros raíz. En cualquier caso, el resultado es el mismo: Ejecutar un sistema Linux completamente desde disquete.

El ejemplo más claro de un disco de arranque/raíz son los discos de arranque Slackware[5]. Estos disquetes contienen un núcleo arrancable y un sistema de ficheros raíz, todo en disquete. Están diseñados para usarse en la instalación de la distribución Slackware, pero vienen muy bien cuando hay que hacer mantenimiento del sistema.

El disco de arranque/raíz de H.J Lu, disponible en `/pub/Linux/GCC/rootdisk` en `sunsite.unc.edu`, es otro ejemplo de este tipo de discos de mantenimiento. O, si se es ambicioso, se puede crear uno su propio disco. En muchos casos, sin embargo, la utilización de un disco de arranque/raíz prefabricado es mucho más simple y probablemente será más completo.

La utilización de un disco de arranque/raíz es muy simple. Tan sólo arranque el sistema con el

[5] Véase la sección 2.1.1 para la información sobre cómo obtener ésta desde Internet. Para este procedimiento, no se necesita obtener la versión completa de Slackware —solo los disquetes de arranque y raíz.

disco, y haga login como `root` (normalmente sin password). Para poder acceder a los ficheros del disco duro, se necesitará montar el sistema de ficheros a mano. Por ejemplo, el comando

`#` *mount -t ext2 /dev/hda2 /mnt*

montará un sistema de ficheros ext2fs existente en `/dev/hda2` bajo `/mnt`. Recuerde que `/` es ahora el propio disco de arranque/raíz; se necesitará montar los sistemas de ficheros de su disco duro bajo algún directorio para poder acceder a los ficheros. Por lo tanto, el fichero `/etc/passwd` de su disco duro es ahora `/mnt/etc/passwd` si se montó el sistema de ficheros raíz bajo `/mnt`.

4.11.2 Arreglando la password de root

Si se olvida de la password de root, no hay problema. Sólo hay que arrancar del disco de arranque/raíz, montar su sistema de ficheros raíz en `/mnt`, y eliminar el campo de password de `/root` en `/mnt/etc/passwd`, como por ejemplo:

```
root::0:0:root:/:/bin/sh
```

Ahora `root` no tiene password; al rearrancar desde el disco duro debería ser capaz de hacer login como `root` y poner la password que desee utilizando `passwd`.

¿No le gustó el haber aprendido a utilizar `vi`? En su disco de arranque/raíz probablemente no estarán disponibles otros editores como pueda ser Emacs, pero `vi` debería estarlo.

4.11.3 Arreglando sistemas de ficheros corrompidos

Si se corrompiese de alguna forma el sistema de ficheros, se puede ejecutar `e2fsck` (si se usa el sistema de ficheros ext2fs, por supuesto) para corregir desde disquete cualquier dato dañado en el sistema de ficheros. Otros tipos de sistemas de ficheros utilizan diferentes formas de comando `fsck`; véase la Sección 4.8 para más detalles.

Cuando se comprueba el sistema de ficheros desde disquete, es mejor que no esté montado.

Una causa común de daño en un sistema de ficheros es la corrupción del super bloque. El *super bloque* es la "cabecera" del sistema de ficheros que contiene información acerca del estado del sistema de ficheros, tamaño, bloques libres, y demás. Si se corrompe el super bloque (por ejemplo, escribiendo accidentalmente datos directamente a la partición del sistema de ficheros), el sistema no puede reconocer nada del sistema de ficheros. Cualquier intento de montar el sistema de ficheros fallará y `e2fsck` no será capaz de arreglar el problema.

Felizmente, el tipo de sistema de ficheros *ext2fs* salva copias del super bloque en los límites de "grupos de bloques" en el disco —normalmente cada 8K bloques. Para poder decirle al `e2fsck` que utilice una copia del super bloque, se puede utilizar un comando tal que

`#` *e2fsck -b 8193* ⟨*partición*⟩

donde ⟨*partición*⟩ es la partición en la que reside el sistema de ficheros. La opción `-b 8193` le dice al `e2fsck` que utilice la copia del super bloque almacenada en el bloque 8193 del sistema de ficheros.

4.11.4 Recuperando ficheros perdidos

Si accidentalmente se borrasen ficheros importantes del sistema no habría forma de “desborrarlos”. Sin embargo, se pueden copiar los ficheros relevantes desde el disquete al disco duro. Por ejemplo, si se hubiese borrado `/bin/login` de su sistema (que le permite hacer login), simplemente arranque del disquete de arranque/raíz, monte el sistema de ficheros raíz en `/mnt`, y use el comando

```
# cp -a /bin/login /mnt/bin/login
```

La opción `-a` le dice a `cp` que conserve los permisos en los ficheros que se están copiando.

Por supuesto, si los ficheros que se borraron no fueses ficheros esenciales del sistema que tengan contrapartidas en el disquete de arranque/raíz, se habrá acabado la suerte. Si se hicieron copias de seguridad, siempre se podrá recuperar de ellas.

4.11.5 Arreglando librerías corrompidas

Si accidentalmente se llegasen a corromper las librerías de enlaces simbólicos en `/lib`, es más que seguro que comandos que dependan de estas librerías no vuelvan a funcionar (Véase la sección 4.7.2). La solución más simple es arrancar del disquete de arranque/raíz, montar el sistema de ficheros raíz y arreglar las librerías en `/mnt/lib`.

Capítulo 5

Características avanzadas

En este capítulo nos introduciremos en ciertas características avanzadas, pero interesantes de Linux. Se asume que ya se tiene cierta experiencia en Unix, y se entiende la información contenida en los capítulos precedentes.

El aspecto más importante que distingue a Linux de otras implementaciones de Unix es su diseño abierto a nuevas ideas y su filosofía. Linux no fue desarrollado por un pequeño equipo de programadores liderado por una empresa con un objetivo comercial. Fue desarrollado por un grupo creciente de "hackers" quienes aportaban todo lo que sabían en desarrollo de Unix. Por ello, es enorme la cantidad de software adicional disponible.

5.1 El sistema X Window

X Windows es un enorme entorno gráfico para sistemas Unix, potente y también complejo. Originalmente fue desarrollado por el MIT, y desde entonces los desarrolladores de sistemas comerciales han convertido a las X en un estándar de las plataformas Unix. Prácticamente todas las estaciones de trabajo Unix del mundo utilizan X Window en alguna de sus variantes.

Existe, desarrollada por un equipo de programadores inicialmente liderados por David Wexelblat[1], una implementación gratuita de las X Windows versión 11, *release* 6 (X11R6) para sistemas Unix con 80386/80486/Pentium. Esta implementación, llamada XFree86[2], se encuentra disponible en Unix x86 como System V/386, 386BSD y por supuesto Linux. Incluye todos los binarios necesarios, ficheros de soporte, librerías y utilidades.

Configurar y utilizar a fondo las X Windows se sale de los objetivos de este libro. Le animamos a leer *The X Window System: A User's Guide*—vea el apéndice A para más información sobre el libro. En esta sección, le daremos una descripción de cómo instalar y configurar paso a paso XFree86 en Linux, pero necesitará completar la información con otra disponible en la documentación propia de

[1]Se puede contactar con David en Internet mediante la dirección `dwex@XFree86.org`

[2]XFree86 es marca registrada de The XFree86 Project, Inc.

XFree86 (documentación que comentaremos más adelante). Otra fuente interesante de información es la *XFree86 HOWTO*.

5.1.1 Requisitos de hardware

Al igual que la versión XFree86 3.1 de Septiembre de 1994, están soportados los siguientes "chipsets". La documentación que se incluye con su adaptador de video debería especificar cuál es el "chipset" que utiliza. Si está a punto de comprarse una nueva tarjeta gráfica, conviene pedir al vendedor el modelo exacto de tarjeta y "chipset" de video. Esto supondrá, casi siempre, recurrir al servicio técnico, pero normalmente los vendedores no pondrán inconvenientes. La mayor parte de los vendedores de PCs se limitan a decir que sus ordenadores tienen tarjetas gráficas "SVGA estándar" que "tiene que funcionar" en su sistema. Explíqueles que su software (nombrando a Linux y XFree86, por supuesto) no soporta todos los "chipsets" y que usted necesita por eso información más detallada.

De todas formas, puede determinar su "chipset" ejecutando el programa `SuperProbe` incluído con la distribución de XFree86. De esto hablamos más abajo.

Están soportados los siguientes "chipsets" SVGA:

- Tseng ET3000, ET4000AX, ET4000/W32
- Western Digital/Paradise PVGA1
- Western Digital WD90C00, WD90C10, WD90C11, WD90C24, WD90C30, WD90C31, WD90C33
- Genoa GVGA
- Trident TVGA8800CS, TVGA8900B, TVGA8900C, TVGA8900CL, TVGA9000, TVGA9000i, TVGA9100B, TVGA9200CX, TVGA9320, TVGA9400CX, TVGA9420
- ATI 18800, 18800-1, 28800-2, 28800-4, 28800-5, 28800-6, 68800-3, 68800-6, 68800AX, 68800LX, 88800
- NCR 77C22, 77C22E, 77C22E+
- Cirrus Logic CLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428, CLGD5429, CLGD5430, CLGD5434, CLGD6205, CLGD6215, CLGD6225, CLGD6235, CLGD6420
- Compaq AVGA
- OAK OTI067, OTI077
- Avance Logic AL2101
- MX MX68000, MX680010
- Video 7/Headland Technologies HT216-32

También están soportados los siguientes, con acelerador:

- 8514/A (y clónicos)
- ATI Mach8, Mach32
- Cirrus CLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428, CLGD5429, CLGD5430, CLGD5434, CLGD6205, CLGD6215, CLGD6225, CLGD6235
- S3 86C911, 86C924, 86C801, 86C805, 86C805i, 86C928, 86C864, 86C964
- Western Digital WD90C31, WD90C33
- Weitek P9000
- IIT AGX-014, AGX-015, AGX-016
- Tseng ET4000/W32, ET4000/W32i, ET4000/W32p

Todas las tarjetas gráficas que usan estos "chipsets" están soportadas, incluso para bus local Vesa o PCI.

El soporte incluye los modos monocromáticos y de 256 colores, con la excepción de Avance Logic, MX y Video 7, que están soportadas solo en 256 colores. Si su tarjeta gráfica tiene suficiente memoria DRAM, la mayor parte de los "chipsets" anteriores también están soportados para los modos de 16 y 32 bits por pixel (en concreto, algunas Mach32, P9000, S3 y Cirrus). Lo habitual, sin embargo, es usar 8 bits por pixel, es decir, 256 colores.

El servidor monocromático también soporta las tarjetas VGA genéricas, la Hercules, la Hyundai HGC1280, Sigma LaserView y Apollo. Para la Compaq AVGA, solo están soportadas 64Kb de memoria de vídeo en el servidor monocromático, y la GVGA no ha sido probada con más de 64 Kb.

Indudablemente, esta lista seguirá creciendo conforme pase el tiempo. Las notas que encontrará junto a la versión actual de XFree86 contendrán probablemente una completa lista de "chipsets" soportados en ese momento.

Un problema que se encuentran con frecuencia los desarrolladores de XFree86 es que algunos fabricantes de tarjetas gráficas utilizan mecanismos no estándares para determinar las frecuencias de reloj a usar en la tarjeta. Algunos de esos fabricante tampoco editan especificaciones que describan cómo programar la tarjeta, o exigen a los desarrolladores que firmen un documento de no divulgación para poder obtener la información. Esto, obviamente, dificulta la libre distribución del software de XFree86. Durante mucho tiempo esto ha impedido trabajar con las tarjetas Diamond, pero desde la versión 3.1 de XFree86, Diamond ya ha comenzado a colaborar con el equipo de desarrollo para la realización de `drivers` para esas tarjetas.

La configuración que se sugiere para ejecutar XFree86 con Linux es una máquina 486 con 8 megabytes de RAM por lo menos, y una tarjeta gráfica con un "chipset" de los listados antes. Para obtener un rendimiento óptimo, se sugiere utilizar una tarjeta acelerada, como las basadas en el "chipset" S3. Usted debe comprobar sobre la documentación de XFree86 que su tarjeta está soportada antes de proceder al gasto que supone el nuevo hardware. Se han comparado rendimientos de diversos modelos de tarjetas, y sus resultados son enviados periódicamente a los grupos de `news` USENET `comp.windows.x.i386unix` y `comp.os.linux.misc`.

Como nota adicional, mi sistema personal con Linux es un 486DX2-66 con 20 megabytes de RAM y equipado con una tarjeta BLV con el “chipset” S3-864 y 2 megabytes de DRAM. He tenido rendimientos en X superiores a los de una Sun Sparc IPX. El sistema Linux resulta unas siete veces más rápido que el Sparc IPX (para el curioso, XFree86-3.1 bajo Linux, con esta tarjeta gráfica, consigue unos 171,000 `xstones` mientras que la Sparc IPX consigue solo 24,000). En general, XFree86 en un sistema Linux con una tarjeta SVGA acelerada le dará mucho más rendimiento que el que se encuentra en estaciones de trabajo Unix comerciales (que suelen emplear solo “framebuffers”).

Su máquina necesitará al menos 4 megabytes de memoria física en RAM y 16 de memoria virtual (por ejemplo, 8 megabytes en RAM y otros 8 en swap). Recuerde que cuanta más RAM tenga, menos tendrá el sistema que acudir al disco duro cuando no hay memoria suficiente. Dado que el *swapping* es inherentemente lento, tener 8 megabytes de RAM o más es necesario si quiere correr XFree86 de forma confortable. Un sistema con 4 megabytes de RAM correrá *mucho* más lento (hasta 10 veces) que uno con 8 megabytes o más.

5.1.2 Instalación de XFree86

La distribución en binario de XFree86 para Linux puede encontrarse en muchos servidores de FTP. En `sunsite.unc.edu` se encuentra en el directorio `/pub/Linux/X11`. (En el momento de escribir este libro, la versión era la 3.1; aunque periodicamente aparecen nuevas versiones).

Es común que usted haya obtenido XFree86 como parte de una distribución de Linux, en cuyo caso no necesitará obtener el software del servidor anterior.

Si en cambio los obtiene por esta vía, la siguiente tabla le informará de los ficheros de los que consta la distribución XFree86-3.1.

Se requiere uno de los siguientes servidores:

Fichero	Descripción
`XF86-3.1-8514.tar.gz`	Servidor para tarjetas tipo 8514.
`XF86-3.1-AGX.tar.gz`	Servidor para tarjetas tipo AGX.
`XF86-3.1-Mach32.tar.gz`	Servidor para tarjetas tipo Mach32.
`XF86-3.1-Mach8.tar.gz`	Servidor para tarjetas tipo Mach8.
`XF86-3.1-Mono.tar.gz`	Servidor monocromo.
`XF86-3.1-P9000.tar.gz`	Servidor para tarjetas tipo P9000.
`XF86-3.1-S3.tar.gz`	Servidor para tarjetas tipo S3.
`XF86-3.1-SVGA.tar.gz`	Servidor para tarjetas SVGA.
`XF86-3.1-VGA16.tar.gz`	Servidor para VGA en 16 colores.
`XF86-3.1-W32.tar.gz`	Servidor para tarjetas tipo ET4000/W32.

Además, son necesarios todos los ficheros siguientes:

Fichero	Descripción
XF86-3.1-bin.tar.gz	Binarios de X11R6.
XF86-3.1-cfg.tar.gz	Ficheros de configuración para xdm, xinit y fs.
XF86-3.1-doc.tar.gz	Documentación y páginas de manual.
XF86-3.1-inc.tar.gz	Ficheros *include*.
XF86-3.1-lib.tar.gz	Ficheros de soporte y librerías compartidas.
XF86-3.1-fnt.tar.gz	Fuentes básicas.

Los siguientes ficheros son opcionales:

Fichero	Descripción
XF86-3.1-ctrb.tar.gz	Diversas contribuciones.
XF86-3.1-extra.tar.gz	Otros servidores y binarios.
XF86-3.1-lkit.tar.gz	Kit para personalización de servidores.
XF86-3.1-fnt75.tar.gz	Fuentes de 75 ppp.
XF86-3.1-fnt100.tar.gz	Fuentes de 100 ppp.
XF86-3.1-fntbig.tar.gz	Otras fuentes (Kanji...).
XF86-3.1-fntscl.tar.gz	Fuentes escalables (Speedo, Type1).
XF86-3.1-man.tar.gz	Páginas de manual.
XF86-3.1-pex.tar.gz	Binarios PEX y librerías.
XF86-3.1-slib.tar.gz	Librerías estáticas de X.
XF86-3.1-usrbin.tar.gz	*Daemons* que residen en /usr/bin.
XF86-3.1-xdmshdw.tar.gz	Versión de xdm para *shadow password*.

El directorio de XFree86 debería contener los ficheros README y otras notas acerca de la instalación de la versión actual.

Todo lo que se necesita para instalar XFree86 es obtener los ficheros anteriores, crear el directorio /usr/X11R6 (como root), y desempaquetar los ficheros desde ese directorio con un comando como el siguiente:

```
# gzip -dc XF86-3.1-bin.tar.gz | tar xfB -
```

Recuerde que esos ficheros tar se empaquetan con directorios relativos a /usr/X11R6, con lo que es imprescindible desempaquetarlos en el lugar indicado.

Después de desempaquetar los ficheros, necesitará hacer un enlace del nombre /usr/X11R6/bin/X al servidor que vaya a utilizar. Por ejemplo, si desea usar el servidor SVGA color, /usr/bin/X11/X debe ser un enlace a /usr/X11R6/bin/XF86_SVGA. Si desea utilizar el servidor monocromo en su lugar, el enlace apuntará a XF86_MONO con el comando

```
# ln -sf /usr/X11R6/bin/XF86_MONO /usr/X11R6/bin/X
```

El mecanismo es similar para cualquier otro servidor que quiera usar.

Si no está seguro de qué servidor debe usar, o no conoce el "chipset" de su tarjeta gráfica, puede ejecutar SuperProbe, un programa del directorio /usr/X11R6/bin (incluido en el fichero

XF86-3.1-bin), que intentará identificar su "chipset" y otras informaciones. Anote su salida para posterior referencia.

Necesita asegurarse de que /usr/X11R6/bin se encuentra en el *path*. Esto puede hacerse editando el fichero /etc/profile o /etc/csh.login (según qué *shell* utilice). O simplemente puede añadir el directorio a su path personal modificando su propio .bashrc o .cshrc.

Además, hay que asegurarse que /usr/X11R6/lib es localizable por ld.so, el enlazador dinámico. Para ello, añada la línea

```
/usr/X11R6/lib
```

al fichero /etc/ld.so.conf, y ejecute /sbin/ldconfig, como root.

5.1.3 Configuración de XFree86

Poner a punto XFree86 no suele ser difícil. Sin embargo, si va a usar manejadores de "chipsets" aun en desarrollo o desea obtener el mejor rendimiento o resolución de una tarjeta aceleradora, en configurar a la perfección XFree86 puede tardar bastante.

En esta sección vamos a describir cómo crear y editar el fichero XF86Config, que configura al servidor. En muchos casos lo mejor es comenzar con una configuración "básica", de baja resolución, como 640x480 puntos, que es soportada por todas las tarjetas y monitores. Una vez comprobado su funcionamiento a baja resolución, puede pasar a modificar la configuración para explotar toda la capacidad de su tarjeta de video.

Además de la información que nosotros le damos, debería leerse la siguiente documentación:

- La documentación propia de XFree86 en /usr/X11R6/lib/X11/doc (viene con el paquete XFree86-3.1-doc). Debe prestar especial atención al fichero README.Config, que es un tutorial sobre la configuración de XWindows.
- Diferentes "chipsets" de video tendrán sus propios ficheros README en el directorio anterior (por ejemplo, README.Cirrus). Léase el que le corresponda.
- Las páginas de manual para XFree86.
- Las páginas de manual para XF86Config.
- Las páginas de manual para el servidor concreto que utilice (como XF86_SVGA o XF86_S3).

El fichero de configuración principal debe ser /usr/X11R6/lib/X11/XF86Config. Este fichero contiene información acerca del ratón, parámetros de la tarjeta gráfica, etc. El fichero XF86Config.ed es un ejemplo que incluye la distribución. Copie este fichero como XF86Config y modifíquelo para su sistema.

Las páginas de manual para XF86Config explican el formato del fichero con detalles. Lea ahora esas páginas si aun no lo ha hecho.

Le vamos a presentar un ejemplo de `XF86Config`. No tiene por qué coincidir con el ejemplo que incluye la distribución, pero en todo caso tendrán estructura parecida.

◇ Observe que el formato del fichero puede cambiar algo entre versiones de XFree86. La información que le damos solo vale para la versión 3.1.

◇ Además, no debe limitarse a copiar este fichero e intentar probarlo en su sistema. Si lo hace, y, por error, envía a su monitor una frecuencia demasiado alta para él puede producir una avería. No sería el primer caso. Esto ha ocurrido especialmente con monitores de frecuencia fija. En definitiva, asegúrese de que su fichero `XF86Config` se corresponde con su hardware antes de intentar utilizarlo.

Cada sección del siguiente ejemplo de `XF86Config` se enmarca en un par de líneas `Section` ⟨*section-name*⟩... `EndSection`. La primera parte del fichero es la sección de `Ficheros`, que tendrá el siguiente aspecto:

```
Section "Files"
    RgbPath     "/usr/X11R6/lib/X11/rgb"
    FontPath    "/usr/X11R6/lib/X11/fonts/misc/"
    FontPath    "/usr/X11R6/lib/X11/fonts/75dpi/"
EndSection
```

La línea `RgbPath` pone la ruta de acceso a la base de datos de colores RGB para X11R6, y cada línea `FontPath` pone rutas de acceso a directorios con fuentes X11. Por lo general no tendrá que modificar estas líneas. Tan sólo debe comprobar que en los directorios correspondientes se encuentran los ficheros implicados (fuentes, etc).

La siguiente sección es `ServerFlags`, que especifica algunos “flags” generales para el servidor. Por lo general la sección suele ir vacía.

```
Section "ServerFlags"
# Si quita el comentario siguiente, la llegada de una se˜nal al
# provocara el volcado de su memoria. Esto puede dejar la consola en
# un estado inutilizable pero el volcado puede servirle en la depuracion.
#    NoTrapSignals

# Quitando este comentario se desactiva la secuencia <Crtl><Alt><BS>
#    DontZap
EndSection
```

La siguiente sección es `Keyboard`. Su significado es claro.

```
Section "Keyboard"
    Protocol    "Standard"
    AutoRepeat  500 5
    ServerNumLock
EndSection
```

Existen otras opciones. Vea su fichero `XF86Config` si desea modificar las opciones de teclado. Las que trae por defecto suelen funcionar bien en casi todos los sistemas.

La siguiente sección es `Pointer`, y especifica los parámetros para el ratón.

```
Section "Pointer"

    Protocol    "MouseSystems"
    Device      "/dev/mouse"

# Los dos parametros siguientes solo son necesarios en algunos Logitech.
#    BaudRate   9600
#    SampleRate 150

# La siguiente opcion es util para ratones Microsoft de dos botones.
#    Emulate3Buttons

# La siguiente opcion es util para algunos ratones con tres botones Logitech.
#    ChordMiddle

EndSection
```

Le pueden interesar las opciones `Protocol` y `Device`. `Protocol` especifica el *protocolo* que utiliza su ratón (no se trata de su marca o modelo). Bajo Linux los protocolos válidos son:

- `BusMouse`
- `Logitech`
- `Microsoft`
- `MMSeries`
- `Mouseman`
- `MouseSystems`
- `PS/2`
- `MMHitTab`

`BusMouse` es el que se usa para ratón `Logitech` en bus. Observar que los ratones antiguos de Logitech usan `Logitech` pero los más modernos usan `Microsoft` o `Mouseman`. Este es un caso en el que el protocolo no coincide con la marca.

`Device` especifica el dispositivo donde está el ratón. En muchos sistemas Linux se encuentra en `/dev/mouse`, que suele ser un enlace al dispositivo verdadero (puerto serie como `/dev/cua0`, etc). Por supuesto, debe asegurarse de que el dispositivo que corresponda esté en `/dev`.

La siguiente sección es `Monitor`, que especifica las características de su pantalla TRC. La sección `Monitor` de `XF86Config` puede no ser única, como sucede con otras secciones. Es útil si hay varios monitores conectados al sistema, o si se quiere usar el mismo `XF86Config` en diferentes configuraciones hardware. Por lo general, sin embargo, solo necesitará una sección `Monitor`.

```
Section "Monitor"

    Identifier  "CTX 5468 NI"

    # !! Estos valores son validos solo para un monitor "CTX 5468 NI"
    # No intente usarlo en su monitor si no es este mismo.

    Bandwidth    60
    HorizSync    30-38,47-50
    VertRefresh  50-90

    # Modos: Nombre     reloj    horiz               vert

    ModeLine "640x480"  25       640 664 760 800     480 491 493 525
    ModeLine "800x600"  36       800 824 896 1024    600 601 603 625
    ModeLine "1024x768" 65       1024 1088 1200 1328 768 783 789 818

EndSection
```

La línea `Identifier` se usa para dar a la sección un nombre. Puede ser cualquier cadena de caracteres y se usará después para referirse a ese monitor.

`HorizSync` es el valor de frecuencia de sincronismo horizontal válida, en kHz. Si tiene un monitor "multisync", puede tratarse de un rango (o diversos rangos separados por comas). Si se trata de un monitor de frecuencia fija, será un conjunto discreto de valores como el siguiente:

```
    HorizSync    31.5, 35.2, 37.9, 35.5, 48.95
```

El manual de su monitor debería incluir estos datos en las especificaciones técnicas. Si no tiene esta información, contacte con el vendedor o fabricante para obtenerla. Por supuesto, hay otras fuentes de información.

`VertRefresh` especifica las frecuencias de refresco vertical válidas (o sincronismo vertical), en Hz. Como `HorizSync` puede ser uno o más rangos o valores discretos. También deben indicarse en su manual.

Los dos parámetros anteriores solo se utilizan para comprobar que los modos gráficos pedidos van a ser tolerados por su monitor. Esos modos se especifican en las entradas `Modeline` que tienen el formato siguiente:

`ModeLine` ⟨*nombre*⟩ ⟨*reloj*⟩ ⟨*valores-horiz*⟩ ⟨*valores-vert*⟩

⟨*nombre*⟩ es un nombre cualquiera que le damos al modo. ⟨*reloj*⟩ es la frecuencia de reloj o "dot clock" asociada al modo. Se suele especificar siempre en Mhz, y es la frecuencia de envío de pixels al monitor para esa resolución. ⟨*valores-horiz*⟩ y ⟨*valores-vert*⟩ son cuatro números que especifican cuándo el haz de electrones del monitor debe dispararse y cuándo se envían los pulsos de sincronismo.

Se preguntará cómo determinar esos valores para su monitor. El fichero `Videomodes.doc`, incluido en la distribución de XFree86 describe en detalle cómo obtener los valores requeridos. En primer lugar, ⟨*reloj*⟩ debe corresponderse con un valor de los que pueda producir su tarjeta de video. Por lo tanto solo usará modos con valores de reloj soportados.

En la distribución se incluyen dos ficheros que pueden darle los datos para `Modeline` de su monitor. Estos ficheros son `modeDB.txt` y `Monitors`, ambos en `/usr/X11R6/lib/X11/doc`.

Debería comenzar con valores para el estándar VESA, que la mayor parte de los monitores soportan. Estos valores están incluidos en `modeDB.txt`. En este fichero encontrará entradas como

```
# 640x480@60Hz Non-Interlaced mode
# Horizontal Sync = 31.5kHz
# Timing: H=(0.95us, 3.81us, 1.59us), V=(0.35ms, 0.064ms, 1.02ms)
#
# name        clock   horizontal timing     vertical timing      flags
 "640x480"     25.175  640  664  760  800    480  491  493  525
```

Esta es una temporización estándar VESA para modo 640x480. Utiliza un reloj de 25.175, que debe ser soportado por la tarjeta (esto lo veremos luego). Para incluir esta entrada en el fichero `XF86Config` usaría la línea

```
ModeLine "640x480" 25.175  640 664 760 800  480 491 493 525
```

Observe que el argumento ⟨*nombre*⟩ (en este caso, `"640x480"`) es una cadena de caracteres arbitraria, aunque se suele seguir la convención de indicar la resolución del modo.

Para cada `Modeline` indicado, el servidor comprobará que sus valores caen dentro de los rangos permitidos de frecuencias (especificados antes), anulando el modo si no ocurre así. Principalmente, el valor de reloj no debe ser mayor que el indicado en `Bandwidth`. Sin embargo, usar valores ligeramente superiores no suele dar problemas.

Si los tiempos estándares en VESA no le funcionan correctamente (lo sabrá cuando después los pruebe) mire los ficheros `modeDB.txt` y `Monitors`, que incluyen valores específicos para muchos modelos de monitores. Puede crear entradas `Modeline` a partir de lo que vea en estos ficheros. Asegúrese de que solo usa valores válidos en su monitor. Vea que muchos monitores de 14 o 15 pulgadas no soportan modos de alta resolución como 1024x768 con señales de reloj lentas. Esto significa que si no encuentra modos de alta resolución para su monitor en esos ficheros, no podrá usar dicha resolución.

Si se encuentra perdido, y no encuentra ningún modo que funcione en su monitor, puede seguir las instrucciones de `VideoModes.doc`, que le ayudarán a calcular las entradas `ModeLine` adecuadas a partir de las especificaciones que se digan en el manual de su monitor. El mismo fichero `VideoModes.doc` explica también otros aspectos del servidor de XFree86 como el formato de la directiva `ModeLine`.

Por último, si obtiene valores para `ModeLine` que son aceptables pero no del todo correcto, puede modificarlos ligeramente para obtener el resultado deseado. Por ejemplo, si cuando ejecuta XFree86 la imagen se desplaza ligeramente, o parece girar, puede seguir las instrucciones de `VideoModes.doc` para intentar corregir esas deficiencias. Además, utilice también los propios controles del monitor. Muchas veces es necesario ajustarlos después de cambiar de modo gráfico. Tener esos controles en la parte frontal del monitor le hará la vida más fácil.

◇ Nunca debe usar valores de tiempo o de `Modeline` pensados para monitores diferentes al suyo. Si lo intenta hacer, y envía al monitor una frecuencia para la que no fué diseñado, puede producirle una avería.

La siguiente sección del fichero `XF86Config` es `Device`, que especifica los parámetros de su tarjeta gráfica. Por ejemplo,

```
Section "Device"
        Identifier "#9 GXE 64"

        # Todavia nada; a~nadiremos los valores despues.

EndSection
```

Esta sección establece las propiedades de su tarjeta gráfica. `Identifier` es una cadena de caracteres usada para identificar su tarjeta, cadena que usará para referirse a las características que en esta sección se detallen.

Inicialmente no necesita incluir nada más a la sección `Device` que la identificación de la tarjeta. Esto es así pues usaremos el servidor X para probar las propiedades de su tarjeta, para incluirlas luego en esta sección. El servidor X es capaz de obtener el "chipset", "RAMDAC" y memoria RAM de su tarjeta.

Antes de hacer esto, sin embargo, necesitamos terminar el fichero `XF86Config`. La siguiente sección es `Screen`, que especifica qué combinación monitor/tarjeta usar para un servidor concreto.

```
Section "Screen"
    Driver     "Accel"
    Device     "#9 GXE 64"
    Monitor    "CTX 5468 NI"
    Subsection "Display"
        Depth      16
        Modes      "1024x768" "800x600" "640x480"
        ViewPort   0 0
        Virtual    1024 768
    EndSubsection
EndSection
```

La línea `Driver` indica qué servidor X utiliza. Los valores válidos son:

- `Accel`: Para servidores `XF86_S3`, `XF86_Mach32`, `XF86_Mach8`, `XF86_8514`, `XF86_P9000`, `XF86_AGX`, y `XF86_W32`;
- `SVGA`: Para el servidor `XF86_SVGA`;
- `VGA16`: Para el servidor `XF86_VGA16`;
- `VGA2`: Para el servidor `XF86_Mono`;

- `Mono`: Para manejadores monocromáticos no VGA que se encuentran en los servidores `XF86_Mono` y `XF86_VGA16`.

Debe asegurarse de que `/usr/X11R6/bin/X` es un enlace simbólico al servidor que va a usar.

La línea `Device` especifica el valor `Identifier` de la sección `Device` que corresponda a la tarjeta de video que se va a usar con este servidor. Antes, hemos creado una sección `Device` con la línea

```
Identifier "#9 GXE 64"
```

Así pues, utilizaremos `"#9 GXE 64"` en la línea `Device` de aquí.

De forma parecida, la línea `Monitor` especifica el nombre de la sección `Monitor` que se va a usar en este servidor. Aquí, `"CTX 5468 NI"` es el identificador `Identifier` utilizado en la sección `Monitor` vista antes.

`Subsection "Display"` define distintas características del XFree86 con su particular combinación de monitor y tarjeta gráfica. El fichero `XF86Config` describe todas las opciones en detalle. La mayoría no son necesarias para que el sistema funcione.

Las opciones que debería conocer son:

- `Depth`. Define el número de bits por pixel (o planos de color), que normalmente valdrá 8 (supone 256 colores). Para el servidor `VGA16` se usará una profundidad de 4, y para el monocromo, una profundidad de 1. Si se utiliza una tarjeta acelerada con suficiente memoria, se puede subir la profundidad a 16, 24 o 32. Si le da problemas, vuelva a profundidad 8 e intente corregir el problema después.

- `Modes`. Es la lista de modos de video que estarán entre los definidos con `ModeLine` para su monitor. Los nombres usados antes para identificarlos (`"1024x768"`, etc) se usarán ahora para expresar los modos de la siguiente manera:

  ```
  Modes    "1024x768" "800x600" "640x480"
  ```

  De esta forma, XFree86 intentará arrancar en el primero de los modos. Una vez que ha arrancado, con las teclas `ctrl`-`alt`-`numeric +` y `ctrl`-`alt`-`numeric -` podrá cambiar a otro modo.

  La mejor forma de empezar, cuando se está configurando XFree86, es usar modos de baja resolución (como 640x480) que suelen funcionar siempre. Una vez que se tenga una configuración básica que va bien, puede pasarse a modificar `XF86Config` para soportar resoluciones mayores.

- `Virtual`. Pone el tamaño del área de trabajo virtual. XFree86 es capaz de usar la memoria sobrante de su tarjeta gráfica para extender el tamaño de su área de trabajo. Cuando mueve el ratón a extremos de la pantalla, verá como la pantalla se desplaza, accediendo así a todo el área virtual. Así pues, si usted trabaja con una resolución inferior, como 800x600 y pone un área de 1024x768 (propia para 1 Mbyte de DRAM y 8 bits por pixel) podrá usar un área de trabajo más grande que la que cabe en pantalla.

  El mecanismo `Virtual` resulta interesante para aprovechar toda la memoria de la tarjeta gráfica, pero es limitado. Si quiere usar un área de trabajo virtual, el autor recomienda usar las facilidades que al respecto dan gestores de ventanas como fvwm. El mecanismo aquí se implementa a base de ocultar ventanas, etc. en lugar de ocupar toda la memoria de la tarjeta. Para más detalle, vea las páginas de manual de fvwm, gestor de ventanas que suelen traer los sistemas Linux por defecto.

- ViewPort. Si usted utiliza la opción Virtual descrita antes, puede usar ViewPort para indicar en qué coordenadas virtuales comenzar XFree86. Se suele usar ViewPort 0 0 que pone el área de trabajo en la esquina superior izquierda. Pero si no se indica se pondrá centrado, algo que quizás le interese más.

Hay muchas otras opciones en esta sección; vea las páginas de manual para XF86Config para más detalle. En la práctica no son necesarias esas otras opciones para que funcione el sistema.

5.1.4 Introducción de los datos de la tarjeta gráfica

Su fichero XF86Config está ya casi listo para usar, aunque aun falta la información de su tarjeta gráfica. Lo que vamos a hacer ahora es usar el servidor X para obtener la información que falta, e introducirla en XF86Config.

A veces se puede encontrar esa información en los ficheros modeDB.txt, AccelCards y Devices, que se encuentran en /usr/X11R6/lib/X11/doc. Además, existen numerosos ficheros README para algunos "chipsets". Debe mirar esos ficheros para buscar su tarjeta y usar los valores que aparezcan en ellos (de reloj, tipo de "chipset" y otros) en la sección correspondiente de XF86Config. Si falta alguna información puede intentar obtenerla con el procedimiento siguiente.

En estos ejemplos vamos a ver cómo configuraríamos el sistema para una tarjeta #9 GXE 64, que usa el servidor XF86_S3. Es la tarjeta que utiliza el autor, aunque el procedimiento es igualmente aplicable a cualquier otra.

Lo primero que debe hacerse es determinar el "chipset" que lleva la tarjeta. Ejecutando /usr/X11R6/bin/SuperProbe lo averiguaremos, aunque también hay que conocer el nombre con el que el servidor X conoce a ese "chipset". Para conseguir esto último, ejecutar:

```
X -showconfig
```

Esto le dará los nombres de "chipsets" que conoce el servidor X (esto también se encuentra en las páginas de manual). Por ejemplo, con el servidor XF86_S3 obtendremos:

```
XFree86 Version 3.1 / X Window System
(protocol Version 11, revision 0, vendor release 6000)
Operating System: Linux
Configured drivers:
  S3: accelerated server for S3 graphics adaptors (Patchlevel 0)
      mmio_928, s3_generic
```

Así pues los nombres de "chipsets" que este servidor tiene son mmio_928 y s3_generic. En la página de manual para XF86_S3 encontrará la descripción de estos "chipsets" junto a algunas tarjetas que los utilizan. Para la #9 GXE 64 el "chipset" es mmio_928.

Si no conoce qué "chipset" usar, el servidor X puede intentar averiguarlo. Para ello, ejecutar

```
X -probeonly > /tmp/x.out 2>&1
```

si usa `bash` como shell. Si usa `csh`, intente:

```
X -probeonly &> /tmp/x.out
```

El comando anterior debe ejecutarse solo cuando no hay otras actividades en el sistema (otros usuarios, procesos, etc.) Este comando obtiene también los valores para los relojes y si el sistema está cargado se calcularán incorrectamente.

Por ejemplo, esto es lo que sale para la tarjeta del autor:

```
XFree86 Version 3.1 / X Window System
(protocol Version 11, revision 0, vendor release 6000)
Operating System: Linux
Configured drivers:
  S3: accelerated server for S3 graphics adaptors (Patchlevel 0)
      mmio_928, s3_generic
algunas líneas borradas ...
(--) S3: card type: 386/486 localbus
(--) S3: chipset:   864 rev. 0
(--) S3: chipset driver: mmio_928
```

Aquí vemos los dos "chipsets" válidos (`mmio_928` y `s3_generic`) del servidor. Este probó ambos y determinó que la tarjeta usa el "chipset" `mmio_928`.

En la sección `Device` del fichero `XF86Config`, añada una línea `Chipset` con el nombre obtenido antes. Por ejemplo,

```
Section "Device"
        # Este es el identificador que le pusimos...
        Identifier "#9 GXE 64"
        # A~nadimos esta linea:
        Chipset "mmio_928"
EndSection
```

Ahora hay que determinar las frecuencias de reloj. Cada una de éstas es simplemente la frecuencia con la que la tarjeta puede enviar píxeles al monitor. Como ya sabemos, estas frecuencias dependen de la resolución usada.

En primer lugar debe echar un vistazo a los ficheros mencionados antes (como `modeDB.txt`) y ver si los relojes de su tarjeta están listados. En su caso verá una lista de 8 ó 16 valores, todos ellos dados en Mhz. Por ejemplo, en `modeDB.txt` encontramos lo siguiente para la tarjeta Cardinal ET4000:

```
# chip    ram   virtual   clocks                          default-mode  flags
 ET4000   1024  1024 768   25  28  38  36  40  45  32   0 "1024x768"
```

Como vemos, los relojes son 25, 28, 38, 36, 40, 45, 32 y 0 MHz.

En la sección `Devices` del fichero `XF86Config` debe añadir una línea `Clocks` como la siguiente (para nuestro ejemplo):

```
Clocks 25 28 38 36 40 45 32 0
```

Esto estará en la sección `Devices`, después de `Chipset`. Observe la importancia del orden de los relojes. No los cambie.

Si no encuentra las frecuencias de reloj para su tarjeta en los ficheros, el servidor X también los puede intentar averiguar. Con el comando ya visto `X -probeonly` obtendrá una línea como la siguiente:

```
(--) S3: clocks:  25.18  28.32  38.02  36.15  40.33  45.32  32.00  00.00
```

Con esta línea podemos hacer la línea `Clocks` en `XF86Config`. Si hay muchos valores y no caben en una línea puede crear más líneas `Clock`, pero respete el orden de todas formas.

Asegúrese de que no hay ninguna línea `Clocks` en la sección `Devices` cuando use el comando `X -probeonly` puesto que en este caso no hará pruebas reales de los relojes, sino que probará solo los valores que haya en el fichero `XF86Config`.

Observe que algunas tarjetas aceleradas tienen chip de reloj programable (vea las páginas de manual de `XF86_Accel` para más detalle. Estas son aplicables a las tarjetas S3, AGX y XGA-2). Este chip viene a permitir que el servidor X diga a la tarjeta qué relojes usar. Si este es el caso, no debe haber una lista de valores válidos de reloj en el fichero `XF86Config`. Si no, la lista de valores de reloj que devuelve `X -probeonly` contendrá solo uno o dos valores discretos de reloj, dejando el resto copiados de los primeros o a cero.

Para placas con reloj programable, deberá usar una línea `ClockChip` en lugar de `Clocks`. Esta línea da el nombre del chip correspondiente a la tarjeta. Esos nombres se encuentran en las páginas de manual. Por ejemplo, en el fichero `README.S3`, veremos que diversas tarjetas S3-864 utilizan un chip "ICD2061A", y deberá existir la línea

```
ClockChip "icd2061a"
```

en lugar de la línea `Clocks` en el fichero `XF86Config`. Como sucede con `Clocks`, la línea formará parte de la sección `Devices` e irá a continuación de `Chipset`.

De manera similar, algunas tarjetas aceleradas requieren que se especifique el tipo de chip RAMDAC del fichero `XF86Config`, con una línea `Ramdac`. Esta opción se describe en `XF86_Accel`. Normalmente, el servidor X ya se ocupa de tomar un valor correcto para el RAMDAC.

Algunas tarjetas gráficas necesitan que se especifiquen diversas opciones en la sección `Devices`. Estas opciones se describen en ficheros como `README.cirrus` o páginas de manual, y se integran en una línea `Option` del fichero `XF86Config`. Por ejemplo, una tarjeta #9 GXE 64 necesita dos opciones:

```
        Option "number_nine"
        Option "dac_8_bit"
```

Normalmente, el servidor X funcionará sin las opciones, pero conviene ponerlas para aumentar el rendimiento de la tarjeta. Hay muchas opciones y muchas particulares de cada tarjeta. Para utilizarlas, mire los ficheros del directorio `/usr/X11R6/lib/X11/doc`.

Así pues, cuando haya acabado, tendrá una sección `Devices` como la siguiente:

```
Section "Device"
        # Solo valida para la tarjeta #9 GXE 64 !
        Identifier "#9 GXE 64"
        Chipset "mmio_928"
        ClockChip "icd2061a"
        Option "number_nine"
        Option "dac_8_bit"
EndSection
```

Observe que en muchos casos se requiere una línea `Clocks` en lugar de `Clockchip`, etc. Lo anterior es sólo un ejemplo.

Hay otras opciones para la sección `Devices`. Los detalles los verá en las páginas de manual correspondientes.

5.1.5 Funcionamiento de XFree86

Con el fichero `XF86Config` terminado, puede probar ya el servidor X. En primer lugar, debe asegurarse de que `/usr/X11R6/bin` está en el *path*.

El comando para iniciar XFree86 es

```
startx
```

Que es un programa que llama a `xinit` (en el caso de que usted haya usado `xinit` para iniciar XWindows en otros sistemas).

Este comando arrancará el servidor X y ejecutará los comandos encontrados en el fichero `.xinitrc` de su directorio *home*. `.xinitrc` es un *shell-script* que contiene los clientes a ejecutar. Si no existe se buscará un equivalente en `/usr/X11R6/lib/X11/xinit/xinitrc`.

Un típico `.xinitrc` sería el siguiente:

```
#!/bin/sh

xterm -fn 7x13bold -geometry 80x32+10+50 &
xterm -fn 9x15bold -geometry 80x34+30-10 &
oclock -geometry 70x70-7+7 &
xsetroot -solid midnightblue &
```

```
exec twm
```

Este script arrancará dos clientes `xterm` y un `oclock`, y pondrá el fondo (root window) en color `midnightblue`. Entonces arrancará el gestor de ventanas `twm`. Observe que `twm` se ejecuta con la sentencia `exec`. Esto quiere decir que el proceso de `xinit` será reemplazado por `twm`. Por lo tanto, al ordenar la terminación de `twm` desde los menús el servidor X también terminará. Para hacerlo pulse el botón 1 del ratón sobre el fondo de la pantalla (no en las ventanas) y verá un menú con la opción de terminar `twm` entre otras.

Asegúrese que el último comando de `.xinitrc` se arranca con `exec`, y no se le pone en *background* (con el "&"). Si no, el servidor X terminará tan pronto se arranquen los clientes de `.xinitrc`.

Otra forma de salir de X es pulsando a la vez `ctrl`-`alt`-`backspace`. Esto detendrá el servidor X y se saldrá del sistema XWindows.

Lo anterior es una configuración de desktop muy sencilla. Hay otros muchos programas X disponibles que pueden especificarse en el fichero `.xinitrc`. Por ejemplo, puede usar `fvwm` en lugar de `twm`, para obtener un desktop virtual, posibilidad de personalizar colores, fuentes, ventanas... En principio puede parecer muy simple el sistema XWindows, aunque con las personalizaciones se descubre lo potente que es.

Si usted es nuevo en X Windows, le sugerimos que consulte libros como *The X Window System: A User's Guide*. Aquí se tratan los temas de X Windows en profundidad, que se completarán con las páginas de manual para `xterm`, `twm`, etc.

5.1.6 Ejecución con problemas

No siempre le saldrá perfecta la primera prueba con el servidor X. Suele venir causado por un error en el fichero `XF86Config`. Normalmente, los valores de reloj del monitor o de la tarjeta han sido mal puestos. Si su pantalla "gira" o los lados están difusos, se trata de un error en los valores para los relojes del monitor. Otra fuente de problemas es especificar mal el "chipset" de la tarjeta de video así como otras opciones de la sección `Device` de `XF86Config`. Compruebe también que el nombre `/usr/X11R6/bin/X` es un enlace simbólico al servidor X que quiera usar.

Si todo lo anterior no le sirve, prueba a iniciar X en modo "bare", o sea, con el comando:

```
X > /tmp/x.out 2>&1
```

A continuación puede *matar* el servidor X (con la combinación de teclas `ctrl`-`alt`-`backspace`) y ver el contenido del fichero `/tmp/x.out`, en el que el servidor X habrá puesto todo tipo de avisos y errores, como los correspondientes a valores de reloj no encontrados en la tarjeta, etc.

El fichero `VideoModes.doc` incluido en la distribución contiene muchas ayudas para calcular los valores para su fichero `XF86Config`.

Recuerde que puede conmutar entre modos de video con las combinaciones de teclas `ctrl`-`alt`-`numeric +` y `ctrl`-`alt`-`numeric -`. Si la máxima resolución no sale bien, es porque alguna parte del fichero `XF86Config` se ha puesto mal.

Compruebe también los mandos de tamaño de imagen del monitor. Muchas veces hay que retocarlos cuando se entra en X. Por ejemplo, si la pantalla se ve desplazada, actúe sobre el mando correspondiente.

El el grupo de *news USENET* `comp.windows.x.i386unix` encontrará más ayuda sobre la utilización de XFree86. Puede ser una buena idea comprobar si en el grupo alguien ha enviado mensajes con configuraciones para un sistema similar al suyo.

5.2 Acceso a ficheros MS-DOS

Si, por alguna extraña razón, necesitara acceder a ficheros MS-DOS, es fácil de hacer en Linux.

La forma habitual de acceder a los ficheros MS-DOS es montar una partición o disquete con un sistema MSDOS, permitiéndole así que acceda a los ficheros a través del sistema de ficheros de Linux. Por ejemplo, si tiene un disquete MSDOS en `/dev/fd0`, el comando

```
# mount -t msdos /dev/fd0 /mnt
```

lo montará en `/mnt`. Vea la sección 4.6.2 para más información acerca del montaje de disquetes.

También puede montar una partición del disco duro con MS-DOS. Si se encuentra en `/dev/hda1`, el comando

```
# mount -t msdos /dev/hda1 /mnt
```

lo montará. Asegúrese de desmontar (`umount`) la partición o el disquete cuando ya no los necesite. También puede pedir a Linux que monte las particiones DOS durante el arranque, para lo que se usa `/etc/fstab`. Vea la sección 4.8 para más detalle. Por ejemplo, la siguiente línea de `/etc/fstab` provocará el montaje de la partición de DOS en el directorio `/dos`:

```
/dev/hda1     /dos     msdos     defaults
```

Las `Mtools` pueden usarse también para acceder a ficheros DOS en disquetes, con comandos `mcd`, `mdir`, `mcopy`, etc. Vea las páginas de manual correspondientes.

Una cosa es acceder a ficheros MS-DOS. Otra es ejecutarlos en Linux. Existe un emulador de DOS bajo Linux en desarrollo, que se puede encontrar fácilmente incluso en algunas distribuciones como SLS. El emulador se porta bien para bastantes aplicaciones, incluyendo Wordperfect. Pero dado lo diferentes que son Linux y MS-DOS, la potencia del emulador siempre estará limitada.

Además, se está trabajando en un emulador de Microsoft Windows para funcionar en X Windows. Vea los grupos de *news* y servidores de FTP para más información.

5.3 Redes con TCP/IP

Linux soporta una implementación completa de los protocolos de red TCP/IP (Transport Control Protocol/Internet Protocol). TCP/IP ha resultado ser hasta ahora el mejor mecanismo de comunicación entre ordenadores de todo el mundo. Con Linux y una tarjeta Ethernet podrá introducir su máquina en una red local o (si se tienen las conexiones apropiadas) a la InterNet, la red TCP/IP de ámbito mundial.

Poner en marcha una pequeña red local de máquinas Unix es fácil. Sólo requiere una tarjeta Ethernet en cada máquina y los cables adecuados así como hardware accesorio (terminadores, etc). Y si su universidad o empresa tiene acceso a la InterNet, podrá insertar su máquina Linux en esta red.

La implementación actual de TCP/IP y los protocolos relacionados para Linux se llama "NET-2". No tiene que ver con la versión NET-2 para BSD. En realidad, se refiere a que es la segunda implementación que se hace para Linux.

NET-2 de Linux soporta también SLIP (*Serial Line Internet Protocol*). SLIP le permitirá acceder a la InterNet con un módem. Si su universidad o empresa proporciona accesos por SLIP, podrá llamar desde su casa al servidor SLIP y conectarse así a la Red. Recíprocamente, si posee en Linux una tarjeta de Red y un módem podrá configurar un servidor SLIP en él.

Para obtener más información de configuración de TCP/IP en Linux, le animamos a que lea el documento *NET-2 HOWTO*, disponible mediante FTP anónimo en `sunsite.unc.edu`. Se trata de una guía completa de configuración, que incluye conexiones mediante Ethernet y SLIP. Otro documento relacionado es el *Ethernet HOWTO*, que se centra en cómo configurar diversos modelos de tarjetas Ethernet. Además, en el Proyecto de Documentación de Linux, al que pertenece este libro, se ha desarrollado otro sobre este tema, *Linux Network Administrator's Guide*. Vea, para más información el apéndice A.

Encontrará más ayuda en el libro *TCP/IP Network Administration*, de Craig Hunt. Contiene información completa acerca del uso y la configuración de TCP/IP en máquinas Unix.

5.3.1 Hardware requerido

Puede utilizar el TCP/IP para Linux sin hardware de red. Así podrá usar el dispositivo "loopback" para conectarse con usted mismo. Aunque parezca poco serio, hay algunos programas que necesitan conexiones de red "loopback" para funcionar.

Sin embargo, si quiere usar Linux en una red TCP/IP Ethernet, necesitará una de las tarjetas soportadas: 3com 3c503, 3c503/16; Novell NE1000, NE2000; Western Digital WD8003, WD8013; Hewlett Packard HP27245, HP27247, HP27250.

Se ha comprobado que también funcionan las siguientes tarjetas clónicas: Clónicas de WD-80x3: LANNET LEC-45; clónicas de NE2000: Alta Combo, Artisoft LANtastic AE-2, Asante Etherpak 2001/2003, D-Link Ethernet II, LTC E-NET/16 P/N 8300-200-002, Network Solutions HE-203, SVEC 4 Dimension Ethernet, 4-Dimension FD0490 EtherBoard 16, D-Link DE-600 y SMC Elite 16.

Sobre este tema encontrará más información en el documento *Ethernet HOWTO*.

Linux también funciona con SLIP, que permite acceder a la red InterNet por teléfono. En este caso, necesitará un módem compatible con el servidor SLIP. Muchos servidores requieren modems de alta velocidad, a 14400 bits por segundo (norma V.32bis).

5.3.2 Configuración de TCP/IP

En esta sección intentaremos explicar cómo configurar una conexión TCP/IP con Ethernet. Notese que este método funcionará en muchos sistemas, pero no siempre. Nuestra explicación debería ser suficiente para aclararle el camino en la configuración de red en su máquina, pero hay además otros detalles que no mencionaremos aquí por su extensión. Le aconsejamos que consulte los documentos *Linux Network Administrators' Guide* y *NET-2 HOWTO* para más información.[3]

En primer lugar, vamos a asumir que su sistema Linux ha sido instalado con el software TCP/IP. Esto incluye clientes como `telnet` y `ftp`, comandos de administración como `ifconfig` y `route` (que suelen estar en `/etc`) y ficheros de configuración de red, como `/etc/hosts`. Los documentos adicionales que hemos mencionado explican cómo instalar todo ese software si aun no lo ha hecho.

También vamos a suponer que el núcleo está compilado con el soporte TCP/IP. Vea la sección 4.7 para informarse de cómo recompilar el núcleo. Para incluir el soporte de red, tendrá que contestar afirmativamente a la pregunta correspondiente que se le hará durante el comando `make config`.

Una vez hecho esto, se deben modificar los ficheros de configuración que usa NET-2. Esta parte suele ser bastante simple, pero suele haber bastante desacuerdo entre las diferentes distribuciones de Linux. Los ficheros pueden estar en `/etc` o en `/usr/etc` o incluso `/usr/etc/inet`. En el peor caso puede usar el comando `find` para localizar los ficheros. A veces los ficheros están también repartidos por varios directorios y no en uno solo.

Lo siguiente es fundamentalmente aplicable a conexiones Ethernet. Si lo que va a usar es SLIP, léase esta sección para ir entendiendo los conceptos y luego vea las instrucciones específicas para SLIP.

5.3.2.1 La configuración de red

Antes de configurar su sistema con TCP/IP necesita conocer cierta información sobre la red. En muchos casos, el administrador local se la proporcionará.

- Dirección IP. Es la dirección única de cada máquina, formada por números separados por puntos. Por ejemplo, 128.253.153.54. El administrador de red le dará este número.

  Si está configurando el modo "loopback" únicamente (esto es, no tiene conexión a la red mediante SLIP o Ethernet) su dirección IP será la 127.0.0.1.

- Máscara de red ("netmask"). Es un número similar a la dirección IP, que determina qué parte de la dirección IP determina el número de sub-red, y qué parte especifica el *host* en la sub-red (si todo esto no lo comprende bien, le sugerimos que lea documentos sobre administración de red).

  La máscara de red es un patrón de bits, que al ser superpuesto a una dirección de la red, le dirá en qué sub-red se encuentra esa dirección. Esto es muy importante para el rutado y, si usted nota que puede comunicar con gente de redes externas pero no con gente de su misma red, es un buen motivo para pensar que tiene mal puesta la máscara.

[3]Algunas de las cosas que aquí se exponen proceden del documento *NET-2 HOWTO* de Terry Dawson y Matt Welsh.

Los administradores de la sub-red habrán seleccionado las máscaras en tiempo de diseño de la red, y serán quienes deban darle esa información. Muchas sub-redes son de "clase C" y usan la máscara 255.255.255.0. Otras sub-redes de "clase B" usan la 255.255.0.0. El código de NET-2 seleccionará automáticamente una máscara que asume que no hay subred.

Todo esto debe aplicarse también a la configuración "loopback". Dado que la dirección "loopback" es siempre la 127.0.0.1, las máscara será la 255.0.0.0. Puede especificarla de forma explícita o dejar que el sistema la ponga por defecto.

- Dirección de red. Es el resultado de la operación lógica AND entre su dirección IP y la máscara. Por ejemplo, si su dirección IP es la 128.253.154.32 y la máscara es 255.255.255.0, su dirección de red será la 128.253.154.0. Con una máscara 255.255.0.0, la dirección sería 128.253.0.0. Si utiliza solo la configuración en "loopback", la dirección de red no existe.

- Dirección de "broadcast". Se utiliza para lanzar paquetes que deben recibir todas las máquinas de la subred. Así pues, si el número de *host* de la subred se obtiene mediante el último octeto de la dirección IP (o sea, la máscara es la 255.255.255.0), su dirección de "broadcast" será su dirección de red operado en OR con 0.0.0.255.

  Por ejemplo, si su número IP es el 128.253.154.32, y la máscara es la 255.255.255.0, la dirección de "broadcast" sería la 128.253.154.255.

  Observe que por motivos *históricos*, algunas subredes están configuradas para usar la dirección de red como dirección de "broadcast". Si tiene dudas, consulte con el administrador de la red. En muchos casos, bastará con copiar la configuración que tengan otras máquinas de la subred y cambiar únicamente el valor IP, por supuesto.

  La dirección "broadcast" tampoco tiene utilidad en una configuración en "loopback".

- Dirección de pasarela. Se trata de la dirección de la máquina que va a ser su pasarela a otras máquinas que no estén en su misma subred. Muchas veces es una dirección IP como la suya, solo que terminada en ".1". Por ejemplo, si su dirección IP es la 128.253.154.32, la de la pasarela podría ser la 128.253.154.1. El administrador se la dirá en cualquier caso.

  En ocasiones puede tener varias pasarelas. Una pasarela o *gateway* es simplemente una máquina que se encuentra a la vez en dos subredes (tiene una dirección IP por cada una), y reparte los paquetes entre ellas. En muchas subredes existe una sola pasarela para comunicarse con las redes externas, pero en otras hay varias, una para cada subred adicional.

  Si su red está aislada de otras, o su máquina se encuentra en configuración "loopback", no necesitará dirección de pasarela.

- Dirección del servidor de nombres. Suele existir un servidor que traduce nombres de máquinas a direcciones IP. El administrador le facilitará su dirección. Puede usted mismo ejecutar en su máquina un servidor de nombres, el programa `named`, en cuyo caso su dirección será la 127.0.0.1. A menos que realmente lo necesite, le recomendamos que procure siempre usar otra máquina distinta. La configuración de `named` es otro tema; y lo primordial aquí es que comunique con la red. Puede tratar estos asuntos más tarde.

  En una configuración "loopback" no es necesario este dato.

Nota para usuarios de SLIP: La información anterior puede necesitarla o no. Cuando use SLIP su dirección IP será determinada de dos formas: bien "estática", lo que significa que será siempre la misma, o bien "dinámica", lo que indica que le será asignada una de las disponibles cada vez que conecte con el servidor SLIP. En la sección sobre SLIP ampliaremos el tema.

NET-2 implementa rutado completo, múltiples rutas, subredes... Lo anterior describe las configuraciones más básicas. Pero la suya puede ser diferente: cuando tenga alguna duda, consulte al administrador de la red, y eche un vistazo a las páginas del manual para `route` e `ifconfig`. La configuración completa de redes TCP/IP supera ampliamente las intenciones de este libro, y con lo anterior sólo pretendemos posibilitar que todo el mundo pueda poner en marcha su sistema en una red ya configurada.

5.3.2.2 Los ficheros de inicio `rc` para trabajo en redes

Los ficheros `rc` son *shell scripts* que se ejecutan durante el arranque del sistema para configurarlo. Son ejecutados por el proceso `init`, y ponen en marcha los *demonios* básicos como `sendmail` o `cron` y además configuran parámetros de la red como la dirección IP y el nombre del *host*. Estos *scripts* se suelen encontrar en `/etc/rc.d` o en `/etc`.

Lo que vamos a hacer aquí es describir los ficheros `rc` que configuran TCP/IP. En Linux son dos: `rc.inet1` y `rc.inet2`. El primero configura parámetros básicos como direcciones IP e información de rutado. El segundo lanza los demonios TCP/IP, principalmente `inetd`, quien se encargará de lanzar cuando haga falta los `telnetd` y demás.

En muchos sistemas se juntan los dos ficheros en uno, el `rc.inet` o `rc.net`. No tiene importancia el nombre, siempre que se ejecuten en el momento adecuado durante el arranque. Para conseguirlo, `init` tiene que saberlo, y para ello existen entradas específicas en el fichero `inittab`. En el peor caso tendría usted que crear las entradas para `rc.inet1` y `rc.inet2` en dicho fichero.

Como hemos dicho, `rc.inet1` configura los parámetros básicos de red. Esto incluye el número IP y dirección de red, y la tabla de rutado. Estas tablas se usan para rutar los datagramas entrantes y salientes de otras máquinas. Lo más simple es tener tres rutas: una para enviar paquetes a su propia máquina, otra para enviarlos a otra máquina de la subred y una tercera para enviarlos a máquinas de otras subredes (a través de una pasarela). Para configurar esto se usan los programas `ifconfig` y `route`, programas que suelen estar en `/etc`.

`ifconfig` se utiliza para configurar el dispositivo interfaz de red con los parámetros que necesita, como la dirección IP, la máscara, dirección de *broadcast* y otros. `route`, por su lado, se utiliza para crear o modificar entradas de la tabla de rutado.

Para muchas configuraciones, el siguiente `rc.inet1` puede valer, aunque, por supuesto, necesitará poner *sus propias* direcciones IP y demás.

```
#!/bin/sh
# /etc/rc.d/rc.inet1 -- Configuracion de TCP/IP

# Configuracion del dispositivo 'loopback'
```

```
HOSTNAME=`hostname`

/etc/ifconfig lo 127.0.0.1      # utiliza por defecto la mascara 255.0.0.0
/etc/route add 127.0.0.1        # una ruta apunta al dispositivo 'loopback'

# Configuracion del dispositivo ethernet. Si solo se usa el 'loopback',
# comentar las lineas siguientes.

# EDITELO con sus propios datos.
IPADDR="128.253.154.32"         # PONGA aqui su direccion IP
NETMASK="255.255.255.0"         # PONGA aqui su mascara de red
NETWORK="128.253.154.0"         # PONGA aqui su direccion de red
BROADCAST="128.253.154.255"     # PONGA aqui su direccion 'broadcast' si
                                # la tiene. Si no, elimine la linea.
GATEWAY="128.253.154.1"         # PONGA aqui su direccion de pasarela

/etc/ifconfig eth0 ${IPADDR} netmask ${NETMASK} broadcast ${BROADCAST}

# Si no tiene direccion de 'broadcast', ponga la anterior linea asi:
# /etc/ifconfig eth0 ${IPADDR} netmask ${NETMASK}

/etc/route add ${NETWORK}

# Lo que sigue solo hace falta si hay pasarela, o sea, si su subred esta
# conectada a otra red.
/etc/route add default gw ${GATEWAY} metric 1

# Fin de la configuracion de ethernet
```

Quizás tenga que estudiarse un poco más el tema para su instalación particular, aunque en la mayor parte de los casos el fichero anterior será suficiente.

`rc.inet2` arranca servidores usados por TCP/IP. El más importante es `inetd`, que queda en segundo plano y escucha por varios puertos de la red. Cuando una máquina intenta conectarse por uno de ellos (por ejemplo, por el de `telnet`), `inetd` envía una copia del servidor correspondiente (en este caso, `in.telnetd`) para que controle el puerto afectado. Esto es mejor que mantener en ejecución todos los servidores de red necesarios (múltiples copias de `in.telnetd`, `in.ftpd` y demás). `inetd` los arranca conforme se van necesitando.

Pero en `rc.inet2` se arrancan también otros *demonios*. `syslogd` se ocupa de acumular los mensajes generados por el núcleo y diversas aplicaciones y tratarlos según diga el fichero `/etc/syslogd.conf` (guardarlos en ficheros, sacarlos por consola,...). `routed` se ocupa de la información de rutado dinámica. Cuando su sistema intenta enviar paquetes a otra red, puede requerir nuevas entradas en las tablas de rutado, que `routed` trata sin necesidad de intervención del usuario.

El ejemplo siguiente solo arranca un número mínimo de servidores. Existen otros que pueden interesarle, como el NFS. Cuando instale TCP/IP en su sistema, es mejor empezar con una configuración sencilla y luego complicarla según sus necesidades.

Observe que en el fichero siguiente se asume que los servidores de red se encuentran en `/etc`, pero pueden estar en otro sitio (en `/sbin`, por ejemplo).

```
#! /bin/sh
# Ejemplo de /etc/rc.d/rc.inet2

# Arrancar syslogd
if [ -f /etc/syslogd ]
then
      /etc/syslogd
fi

# Arrancar inetd
if [ -f /etc/inetd ]
then
      /etc/inetd
fi

# Arrancar routed
if [ -f /etc/routed ]
then
      /etc/routed -q
fi

# Hecho!
```

Otro servidor que puede interesarle es `named`, servidor de nombres, que traducirá nombres (locales) a direcciones IP y viceversa. Si no hay servidor de nombres en su subred o quiere proporcionar nombres nuevos a la misma, necesitará arrancar `named`. Su configuración es más compleja y requiere cierto cuidado y planificación, por lo que le recomendamos consultar bibliografía específica. Sin embargo, no es habitual tener que instalar un servidor de nombres en su sistema.

5.3.2.3 `/etc/hosts`

`/etc/hosts` lleva una lista de direcciones IP y nombres de máquinas que les corresponden. En general, `/etc/hosts` solo contiene entradas para su máquina y quizás alguna otra "importante", como servidores de nombres o pasarelas. Su servidor de nombres local proporciona a otras máquinas traducción automática del nombre de su *host* a su dirección IP.

Por ejemplo, si su máquina es `loomer.vpizza.com` con la dirección IP 128.253.154.32, su `/etc/hosts` sería como este:

```
127.0.0.1             localhost
128.253.154.32        loomer.vpizza.com loomer
```

Si solo usa el "loopback", la única línea necesaria es la que tiene el número 127.0.0.1, añadiendo tras `localhost` el nombre de su máquina.

5.3.2.4 /etc/networks

El fichero `/etc/networks` tiene direcciones de su red y otras, y es usado por el comando `route`. Permite dar nombre a las redes.

Cada subred que quiera añadir a `route` *debe aparecer* en `/etc/networks`.

Por ejemplo,

```
default 0.0.0.0 # rutado por defecto - obligatorio
loopnet 127.0.0.0 # red de 'loopback' - obligatorio
mynet 128.253.154.0 # Ponga aqui su direccion de red
```

5.3.2.5 /etc/host.conf

Este fichero dice a su sistema cómo resolver los nombres de los *hosts*. Debe contener dos líneas:

```
order hosts,bind
multi on
```

Estas líneas indican a los mecanismos de resolución que empiezen buscando en el fichero `/etc/hosts` y luego pregunten al servidor de nombres, si existe. La entrada `multi` permite que para un nombre de máquina haya varias direcciones IP en `/etc/hosts`.

5.3.2.6 /etc/resolv.conf

En este fichero se configura el mecanismo de resolución, especificando la dirección del servidor de nombres y el nombre del dominio de su máquina. El dominio es como un nombre de *host* "mutilado". Por ejemplo, si su máquina se llama `loomer.vpizza.com`, el dominio será `vpizza.com`.

Como fichero `/etc/resolv.conf` de ejemplo, veremos el caso de la máquina `goober.norelco.com` cuyo servidor de nombres es el 127.253.154.5:

```
domain      norelco.com
nameserver  127.253.154.5
```

Con líneas `nameserver` adicionales podrá especificar la existencia de varios servidores de nombres.

5.3.2.7 Ajuste del nombre de su *host*

Para activar el nombre de su *host* debe usar el comando `hostname`. Esto suele hacerse en un fichero como `/etc/rc.local`. Busque en sus ficheros `rc` y busque una llamada a `hostname` como la siguiente:

```
/bin/hostname loomer.vpizza.com
```

Vea que hay que especificar el nombre completo (dominio incluído).

5.3.2.8 Problemas con la configuración

Una vez que haya preparado los ficheros anteriores, habrá que reiniciar Linux para que reconozca las nuevas configuraciones. Luego tendrá que hacer pruebas, para las que lo más indicado es probar aspectos individuales de la red y no tratar de empezar, por ejemplo, lanzando un proceso Mosaic con una conexión X.

Con el comando `netstat` puede ver las tablas de rutado. Esta suele ser la principal fuente de problemas. En la página del manual para este comando encontrará la sintaxis adecuada. Para comprobar que funciona su conexión, le sugerimos probar un cliente como `telnet` para ver si puede conectarse a máquinas de su subred y de otras redes. Esto puede ponerle sobre la pista del problema. Por ejemplo, si puede conectarse a máquinas de otras subredes pero no de la suya propia, puede tratarse de un problema con la máscara de red o las tablas de rutado. Ejecutando `route` como `root` podrá jugar directamente con las entradas de la tabla.

Para hacer estas pruebas de conectividad, utilice direcciones IP y no nombres. Así, si tiene problemas para ejecutar

`$` *telnet shoop.vpizza.com*

la causa puede ser una configuración incorrecta del servidor de nombres. Si funciona usando la dirección IP, se puede casi asegurar que el resto de la configuración está bien hecha. Solo falta que funcione bien el servicio de nombres (probablemente haya que especificar correctamente la dirección del servidor de nombres).

La depuración de configuraciones de red puede ser tarea difícil, y no podemos tratarla aquí. Le sugerimos, si no consigue otra ayuda, que consulte el libro *Linux Network Administrators' Guide* de esta misma serie.

5.3.3 Configuración de SLIP

Con SLIP (Serial Line Internet Protocol) usted puede conectarse a una red TCP/IP mediante una línea serie, como puede ser un módem o una línea dedicada asíncrona. Por supuesto, para usar SLIP tiene que tener acceso a un servidor SLIP. Muchas empresas y universidades proporcionan acceso por poco dinero.

Podemos destacar dos programas relacionados con SLIP: `dip` y `slattach`. Ambos se usan para iniciar una conexión SLIP y por lo tanto son necesarios. No es suficiente con llamar al servidor SLIP con programas como `kermit` y después usar los comandos `ifconfig` y `route`. Esto se debe a que `dip` y `slattach` realizan una llamada especial *ioctl()* para convertir el control de un dispositivo serie a la interfaz de SLIP.

Con `dip` puede llamarse a un servidor SLIP, hacer ciertas negociaciones de entrada con el mismo (intercambio de usuario y password, por ejemplo) y después iniciar la conexión SLIP. Por su lado, `slattach` se limita prácticamente a modificar la línea serie para SLIP, por lo que está indicado para líneas dedicadas que no requieren interacción con el módem o similar. Casi todo el mundo, sin embargo, usa `dip`.

Con `dip` también puede configurar su sistema como servidor SLIP, permitiendo a otras máquinas conectarse a la red a través de su módem y su conexión Ethernet. Vea los manuales en línea de `dip` para más información.

A SLIP se le llama conexión "punto a punto" (point-to-point) pues a ambos lados de la línea existen sólo las dos máquinas involucradas (no como sucede en una ethernet). Esta idea se generaliza y mejora con el protocolo PPP (point-to-point protocol) que también se ha portado a Linux.

Cuando inicia una conexión al servidor SLIP, se le asignará una dirección IP, bien de forma "estática" (su dirección IP es siempre la misma) o "dinámica" (su dirección puede ser diferente de un día para otro). Por lo general, los valores de la dirección y pasarela asignados serán impresos por el servidor SLIP al conectarse. El programa `dip` es capaz de capturar esos valores y configurar su sistema para adaptarse a ellos.

Esencialmente, configurar una conexión SLIP es como configurar la conexión en "loopback" o con ethernet. En las siguientes líneas le mostramos las diferencias. Es importante que vea lo que hemos explicado antes sobre configuración en general, y aplique ahora las modificaciones que le vamos a contar.

5.3.3.1 Conexiones SLIP con asignación de IP estática usando `dip`

Si su servidor SLIP le permite tener la dirección IP estática, lo más adecuado es insertar la dirección y el nombre del *host* en el fichero `/etc/hosts`. Además, debe configurar los ficheros `rc.inet2`, `hosts.conf` y `resolv.conf` como se ha dicho antes.

En el fichero `rc.inet1` también tendrá que introducir cambios, ejecutando `ifconfig` y `route` solo para el dispositivo "loopback", puesto que `dip` hará lo propio con el dispositivo SLIP. Pero si usa `slattach` *sí tendrá* que incluir comandos `ifconfig/route` en `rc.inet1` para el dispositivo SLIP (en breve veremos cómo).

El programa `dip` *debería* configurar sus tablas de rutado para la conexión SLIP. Sin embargo, puede no hacerlo bien, y tendrá que corregirlo ejecutando por su cuenta `ifconfig` o `route` cuando se haya conectado. Quizás le convenga entonces escribirse un `shell script` para hacerlo automáticamente. En muchos casos, la pasarela es el propio servidor SLIP. De todas formas, el comando `dip` puede deducirlo de la información que envía el servidor al conectarse.

Puede que necesite el argumento `pointopoint` en `ifconfig` si ve que `dip` no lo configura bien. Por ejemplo, si la dirección del servidor SLIP es 128.253.154.2 y la suya es 128.253.154.32, el comando a ejecutar (como `root`) podría ser:

```
ifconfig sl0 128.253.154.32 pointopoint 128.253.154.2
```

tras conectar con `dip`. La documentación en línea de este comando le será útil.

Observe que los dispositivos SLIP que se usan en `ifconfig` y `route` son `sl0`, `sl1`, etc. (y no como en ethernet, que es `eth0`, `eth1`, etc.)

En la sección posterior 5.3.4 le explicaremos cómo configurar `dip` para conectarse a un servidor SLIP.

5.3.3.2 Conexiones SLIP con asignación de IP estática usando slattach

Si tiene una línea dedicada o un cable conectado directamente al servidor SLIP, no necesitará usar `dip` para iniciar la conexión. En su lugar puede usar `slattach`.

En este caso, el fichero `/etc/rc.inet1` puede quedar como sigue:

```
#!/bin/sh
IPADDR="128.253.154.32"          # Ponga aqui su direccion IP
REMADDR="128.253.154.2" # Ponga aqui la del servidor de SLIP

# Modifique lo siguiente para su dispositivo serie
slattach -p cslip -s 19200 /dev/ttyS0
/etc/ifconfig sl0 $IPADDR pointopoint $REMADDR up
/etc/route add default gw $REMADDR
```

`slattach` asigna el primer dispositivo SLIP disponible (`sl0`, etc.) a la línea serie especificada.

Observe que el primer parámetro de `slattach` es el protocolo SLIP a utilizar. Actualmente solo valen `slip` y `cslip`. El segundo es un SLIP que incluye compresión de las cabeceras de los datagramas. Por ello su elección habitual será cslip a menos que tenga algún problema con la conexión.

Si hay más de un dispositivo SLIP tendrá que considerar algunas cosas respecto al rutado. Tiene que decidir qué rutas añadir, y esto debe hacerse en función de la configuración de la red a la que se conecte. Le serán de ayuda los libros sobre configuración de TCP/IP, la documentación en línea del comando `route`, etc.

5.3.3.3 Conexiones SLIP con asignación de IP dinámica usando dip

Si el servidor SLIP le asigna dinámicamente las direcciones IP, no sabrá, evidentemente, su dirección IP antes de conectarse, con lo que no puede incluir esa información en `/etc/hosts` (aunque sí incluirá la información de "loopback", 127.0.0.1).

Muchos servidores SLIP envían al terminal la dirección IP y la del propio servidor. Por ejemplo, un servidor SLIP podría decirle esto al conectarse:

```
Your IP address is 128.253.154.44.
Server address is 128.253.154.2.
```

`dip` puede capturar ese texto y configurar así el sistema.

Vea la sección 5.3.3.1 para informarse sobre la configuración de los ficheros de TCP/IP con SLIP. Ahora le indicaremos cómo se configura SLIP para conectarse al servidor SLIP.

5.3.4 Utilización de dip

`dip` puede facilitar el proceso de conexión a un servidor SLIP, pues se ocupará de entrar en el sistema remoto y configurar el dispositivo SLIP según la información recibida del servidor. Este programa es el más indicado a menos que su línea sea dedicada.

Para utilizar `dip` tendrá que escribir un “script” que contendrá comandos para comunicar con el servidor SLIP durante la entrada en el sistema remoto. Por ejemplo, incluirá envío automático de usuario y password al servidor así como lo necesario para asignar la dirección IP.

Lo que sigue es un ejemplo de *script* para asignación dinámica de dirección IP. Para asignación estática puede poner al principio del *script* los valores fijos a `$local` y `$remote` (direcciones IP local y remota, respectivamente). Vea los manuales de `dip` para más información.

```
main:
  # MTU es 'Maximum Transfer Unit' o tama~no maximo de los paquetes
  # transmitidos por el dispositivo SLIP. En muchos servidores este
  # valor debe ser 1500 o 1506. Hable con el administrador de la red
  # si no esta seguro.
  get $mtu 1500

  # Hacer que el rutado de SLIP sea el de su sistema por defecto.
  default

  # Elegir puerto serie y velocidad.
  port cua03
  speed 38400

  # Reiniciar el modem y la linea del terminal. Si le da problemas,
  # comente la linea.
  reset

  # Ponga aqui su cadena de inicio del modem.
  send ATT&C1&D2\\N3&Q5%M3%C1N1W1L1S48=7\r
  wait OK 2
  if $errlvl != 0 goto error
  # Llamar al servidor SLIP (ponga aqui el telefono).
  dial 2546000
  if $errlvl != 0 goto error
  wait CONNECT 60
  if $errlvl != 0 goto error

  # En este punto estaremos conectados. Entrar en el sistema.
login:
  sleep 3
  send \r\n\r\n
  # Esperar el 'prompt' de entrada (login).
  wait login: 10
  if $errlvl != 0 goto error

  # Enviar su nombre de usuario.
  send USERNAME\n
```

```
  # Esperar el 'prompt' de password.
  wait ord: 5
  if $errlvl != 0 goto error

  # Enviar su password.
  send PASSWORD\n

  # Esperar el 'prompt' del servidor que indica que esta preparado.
  wait annex: 30
  if $errlvl != 0 goto error

  # Enviar un comando al servidor para empezar la conexion.
  send slip\n
  wait Annex 30

  # Obtener la direccion IP desde el servidor. El comando 'get...remote'
  # lee un texto de la forma xxx.xxx.xxx.xxx y lo asigna a la variable
  # dada como segundo argumento (aqui es $remote).
  get $remote remote
  if $errlvl != 0 goto error
  wait Your 30

  # Obtener la direccion local IP desde el servidor y asignarla a $local.
  get $local remote
  if $errlvl != 0 goto error

  # Establecer la conexion SLIP.
done:
  print CONNECTED to $remote at $rmtip
  print GATEWAY address $rmtip
  print LOCAL address $local
  mode SLIP
  goto exit
error:
  print SLIP to $remote failed.

exit:
```

`dip` ejecuta automáticamente los programas `ifconfig` y `route` según los valores asignados a `$local` y `$remote`. Aquí, esas variables son asignadas con el comando `get...remote`, que obtiene el texto de la dirección del servidor SLIP y lo asigna a la variable.

Si los comandos `ifconfig` y `route` que `dip` ejecuta no funcionan, siempre puede llamarlos por su cuenta desde un *shell script* tras ejecutar `dip` o modificar las fuentes del propio `dip`. La opción `-v` de `dip` le dará información para depuración generada durante la conexión y le ayudará a averiguar la(s) causa(s) del problema(s).

Ahora, para probar `dip` y abrir la conexión SLIP, escriba un comando como:

```
/etc/dip/dip -v /etc/dip/mychat 2>&1
```

Estando los ficheros de `dip`, incluyendo el *script* `mychat.dip` en `/etc/dip`.

Las explicaciones de esta sección le deberían haber permitido conectarse a la red, bien sea por Ethernet o por SLIP. De nuevo le volvemos a recomendar que consulte un libro sobre configuración de redes TCP/IP, en especial si en la red hay configuraciones especiales de rutado o similar.

5.4 Red con UUCP

UUCP (UNIX-to-UNIX Copy) es un viejo mecanismo usado para transferir información entre sistemas Unix. Mediante UUCP, los sistemas Unix se comunican con otros (vía módem), transfiriendo mensajes de correo, *news*, ficheros y demás. Si no tiene acceso TCP/IP o SLIP, puede usar UUCP para comunicarse con el mundo. Casi todo el software de correo puede ser configurado para usar transferencias UUCP. De hecho, si tiene algún servidor InterNet cercano, puede recibir correo en su sistema de esa red mediante UUCP.

El libro *Linux Network Administrator's Guide* le dará información completa para configurar y utilizar UUCP en Linux. También encontrará información en el documento *UUCP-HOWTO*, que puede obtener por FTP anónimo de `sunsite.unc.edu`. Otra fuente de información sobre UUCP es el libro *Managing UUCP and USENET*, de Tim O'Reilly y Grace Todino. Vea el apéndice A para más información.

5.5 Correo Electrónico

Como casi todos los UNIX, Linux dispone de paquetes de software para tener correo electrónico. Este puede ser tanto local (entre usuarios de su sistema) como remoto (mediante una red TCP/IP o UUCP). El software de *E-Mail* consta normalmente de dos partes: un agente de usuario o *mailer* y un programa de transporte. El agente de usuario es el software que el usuario utiliza para crear mensajes, leerlos, etc. Podemos destacar aquí los programas `elm`, `pine` y `mailx`. El programa de transporte es quien se ocupa de entregar correo tanto remoto como local, conociendo protocolos de comunicaciones y demás. El usuario nunca interactúa directamente con este programa, sino que lo hace a través del agente de usuario. Sin embargo, el administrador del sistema debe conocer cómo funciona el programa de transporte, con el fin de configurarlo según sus necesidades.

En Linux, el más conocido de los programas de transporte es `Smail`. Es fácil de configurar y capaz de enviar tanto correo local como remoto vía UUCP o TCP/IP. En otros sistemas Unix se suele usar con más frecuencia el programa `sendmail`, que es bastante más complicado de configurar, por lo que no se suele usar en Linux.

En el documento *Linux Mail HOWTO* se expone más información sobre el software disponible para correo y cómo configurarlo. Si pretende tener correo remoto, necesitará entender los conceptos de TCP/IP o UUCP (según la red utilizada) (vea las secciones 5.3 y 5.4). Los documentos de UUCP y TCP/IP indicados en el apéndice A también le ayudarán.

Casi todo el software de correo para Linux puede obtenerse mediante FTP anónimo de `sunsite.unc.edu` en el directorio `/pub/Linux/system/Mail`.

5.6 News y USENET

Linux proporciona también todo lo necesario para tratar las *news*. Puede elegir configurar un servidor de news local, que permitirá a los usuarios poner "artículos" a los diversos "grupos" del sistema... en cierto modo, es una forma de discutir. Sin embargo, si tiene acceso a una red UUCP o TCP/IP, podrá participar realmente en USENET, una red de news de ámbito mundial.

En el software de news hay dos partes, el *servidor* y el *cliente*. El servidor de news es el software que controla los grupos de news y se ocupa de enviar los artículos a otras máquinas (si estamos en una red). El cliente, o *lector de news*, es el software que conecta al servidor para permitir que los usuarios lean y escriban artículos.

Hay varios tipos de servidores de news para Linux. Todos siguen un diseño y esquema de protocolos parecido. Principalmente, tenemos los servidores "C News" e "INN". En cuanto a clientes, destacamos `rn` y `tin`. La selección del cliente es cuestión de gustos, y, por supuesto, es independiente del servidor elegido.

Si solo pretende leer y escribir artículos localmente (no como parte de USENET), necesitará un servidor que corra en su sistema, así como el lector para los usuarios. El servidor guardará los artículos en un directorio como `/usr/spool/news`, y el lector se compilará para buscar los artículos en ese directorio.

Sin embargo, si desea tener news en red, tendrá ahora varias opciones más. Para redes basadas en TCP/IP se usa el protocolo NNTP (Network News Transmision Protocol). NNTP permite al cliente leer los artículos a través de la red, desde una máquina remota. NNTP también permite a los servidores enviarse artículos por la red. En esto se basa USENET. Casi todas las empresas y universidades conectadas cuentan con uno o más servidores NNTP para controlar todas las news USENET en ese lugar. Cualquier otra máquina de esa empresa o universidad tendrá un lector de news que accederá al servidor con NNTP. Por ello, solo el servidor NNTP guarda artículos en disco. Los clientes no lo hacen, y siempre tienen que conectar con el servidor para leerlos.

A continuación mostramos algunas situaciones típicas de configuración.

- News locales. No hay conexión a red o no se desea tener news en red. En este caso, hay que ejecutar C News o INN en su máquina, e instalar el lector para leer las news locales.

- Con acceso a red TCP/IP y servidor NNTP. Si existe un servidor NNTP ya configurado, puede leer y escribir artículos desde su máquina Linux instalando un lector basado en NNTP (casi todos los lectores tienen opciones de configuración para leer news en NNTP). Por lo tanto, no necesita preocuparse de instalar el servidor o guardar artículos en su sistema. El lector se ocupará de enviarlos a la red. Por supuesto, necesitará configurar TCP/IP y tener acceso a la red (vea la sección 5.3).

- Tiene acceso a la red TCP/IP pero no hay un servidor NNTP. En este caso, puede instalar un servidor NNTP en su sistema. Además, puede instalarlo para comunicarse con otros servidores NNTP para intercambiar artículos.

- Desea transferir news con UUCP. Si tiene acceso a UUCP (ver la sección 5.4), puede participar en USENET de la misma forma. Necesitará instalar un servidor de news y un lector. Además necesitará configurar su software UUCP para transferir los artículos periódicamente a otra máquina con UUCP. En UUCP no se usa el protocolo NNTP, sino que posee su propio mecanismo para transferir artículos.

El único inconveniente de muchos clientes y servidores de news es que deben ser compilados a mano, es decir, no usan ficheros de configuración, sino que se configuran en el momento de compilarlos.

Muchos programas de news "estándares" (disponibles por FTP anónimo en `ftp.uu.net`, directorio `/news`) podrían no compilarse en Linux. Los parches que hagan falta se encuentran en `sunsite.unc.edu`, directorio `/pub/Linux/system/Mail` (aquí se encuentra también todo el software de correo para Linux). Pueden encontrarse también versiones ya compiladas.

Para más información, léase el documento *Linux News HOWTO* que encontrará en `sunsite.unc.edu`. También encontrará ayuda en el manual *Linux Network Administrator's Guide* de la misma serie de este libro. También puede consultar el libro *Managing UUCP and Usenet*, de Tim O'Reilly y Grace Todino. Una última posibilidad es el documento "How to become a USENET site", disponible en `ftp.uu.net`, directorio `/usenet/news.announce.newusers`.

Apéndice A

Fuentes de Información de Linux

Este apéndice contiene información sobre algunas fuentes de información de Linux, tales como documentos en línea, libros, y demás. Muchos de estos documentos estan disponibles en formato impreso, o electrónicamente desde Internet o sistemas BBS. También muchas de las distribuciones de Linux incluyen esta documentación dentro de ellas, de forma que una vez que haya instalado Linux pueda disponer de estos ficheros en su sistema.

A.1 Documentos en Línea

Estos documentos deberían estar disponibles en cualquiera de los FTP sites de Linux (vea el Apéndice C donde tendrá una lista). Si no tiene acceso directo a FTP, es posible que localice estos documentos en otros servicios en línea (tales como CompuServe, BBSs locales, etc.). Si dispone de acceso a correo Internet, podrá usar el servicio `ftpmail` para recibir estos documentos. Vea el Apéndice C para más información.

En particular, los siguientes documentos se pueden encontrar en `sunsite.unc.edu` en el directorio `/pub/Linux/docs`. Muchos sites hacen copia de este directorio; sin embargo, si no puede localizar un site próximo a Ud., éste es un buen lugar en el que intentarlo.

También puede acceder a los ficheros de Linux y la documentación usando `gopher`. Solo tiene que indicar a su cliente `gopher` que acceda al puerto 70 de `sunsite.unc.edu`, y siga los menús hasta los archivos Linux. Esta es una buena forma de hojear la documentación de Linux de forma interactiva.

La lista de FAQ

La lista de Frequently Asked Questions, o "FAQ", es una lista de preguntas (¡y respuestas!) comunes sobre Linux. Este documento debe proporcionar una fuente de información general sobre Linux, problemas comunes y soluciones, y una lista de otras fuentes de información. Todos los nuevos usuarios de Linux deberían leer este documento. Está a disposición en varios formatos, incluyendo ASCII puro, PostScript, y formato Lout typesetter. El Linux FAQ lo mantiene Ian Jackson, `ijackson@nyx.cs.du.edu`.

El Linux META-FAQ

El META-FAQ es una colección de "metapreguntas" sobre Linux; es decir, fuentes de información sobre el sistema Linux, y otros tópicos generales. Es un buen comienzo para el usuario de Internet que desea encontrar más información sobre el sistema. Lo mantiene Michael K. Johnson, `johnsonm@sunsite.unc.edu`.

El Linux INFO-SHEET

El Linux INFO-SHEET es una presentación técnica del sistema Linux. Ofrece una visión de las características del sistema y el software disponible, y proporciona una lista de otras fuentes de información de Linux. El formato y contenido es similar en naturaleza al META-FAQ; casualmente también lo mantiene Michael K. Johnson.

El Mapa de Software de Linux

El Mapa de Software de Linux es una lista de muchas aplicaciones disponibles en Linux, donde conseguirlas, quien las mantiene, y demás. Está lejos de ser una lista completa—compilar una lista completa de software de Linux sería prácticamente imposible. Sin embargo, incluye muchos de los paquetes más populares de software de Linux. Si no consigue encontrar una aplicación en particular, que se adapte a sus necesidades, el LSM es un buen lugar por el que comenzar. Lo mantiene Lars Wirzenius, `lars.wirzenius@helsinki.fi`.

El Indice de HOWTOs de Linux

Los HOWTOs de Linux son una colección de documentos "How to"[1], cada uno de ellos describiendo un aspecto concreto del sistema Linux. Los mantiene al día Matt Welsh, `mdw@sunsite.unc.edu`. Los Indices de HOWTOs listan los documentos HOWTO que están disponibles (algunos de los cuales se listan aquí abajo).

El HOWTO de Instalación de Linux (Linux Installation HOWTO)

El HOWTO de Instalación de Linux describe como obtener e instalar una distribución de Linux, similar a la información presentada en el Capítulo 2.

El HOWTO de Distribuciónes de Linux (Distribution HOWTO)

Este documento es una lista de las distribuciones de Linux disponibles a través de pedido por correo y FTP anónimo. También incluye información sobre otras bondades y servicios relacionados con Linux. El Apéndice B contiene una lista de vendedores de Linux, la mayoría de los cuales están listados en el *Distribution HOWTO*.

El HOWTO de XFree86 de Linux (Linux XFree86 HOWTO)

Este documento describe como instalar y configurar el software del Sistema X Window para Linux. Ver la sección "5.1" para mayor información acerca del Sistema X Window.

[1] N. del T.: del inglés "Cómo" o "Cómo se hace"

Los HOWTOs de Mail, News y UUCP de Linux
Estos tres documentos HOWTO describen la configuración y puesta a punto del correo electrónico, news y comunicaciones UUCP en un sistema Linux. Dado que estos tres temas están generalmente interconectados, probablemente desee leer estos tres HOWTOs de una misma vez.

El HOWTO de Hardware de Linux (Linux Hardware HOWTO)
Este HOWTO contiene una extensa lista de hardware soportado por Linux. A pesar de que esta lista no está completa, debería ofrecerle una perspectiva general de que dispositivos hardware deben estar soportados por el sistema.

El HOWTO de SCSI de Linux (Linux SCSI HOWTO)
Este HOWTO es una guía completa de la configuración y uso de los dispositivos SCSI bajo Linux, tales como discos duros, unidades de cinta y CD-ROM.

El Linux NET-2-HOWTO
El Linux NET-2-HOWTO describe la instalación, puesta a punto y configuración del software TCP/IP "NET-2" bajo Linux, incluyendo SLIP. Si desea usar TCP/IP en su sitema Linux, debería leer este documento.

El HOWTO de Ethernet de Linux (Linux Ethernet HOWTO)
Muy relacionado con el NET-2-HOWTO, el HOWTO de Ethernet describe los diversos dispositivos Ethernet soportados por Linux, y explica como configurar cada uno de ellos desde el software de TCP/IP de Linux.

El HOWTO de Impresión de Linux (Linux Printing HOWTO)
En este documento se describe como configurar el software de impresión bajo Linux, como por ejemplo `lpr`. La configuración de las impresoras y el software de impresión bajo UNIX puede resultar a veces confuso; este documento siembra algo de luz en la materia.

Otros documentos en-línea
Si hojea el directorio `docs` de cualquier FTP site de Linux, podrá ver otros muchos documentos que no están aquí listados: Una pila de FAQs, trucos de interés, y otra información importante. Esta miscelánea es difícil de clasificar aquí; si no puede ver en la lista de abajo lo que esté buscando, eche un vistazo en los sites de archivos Linux listados en el Apéndice C.

A.2 Manuales del Proyecto de Documentación de Linux

El Proyecto de Documentación de Linux (LDP) está trabajando en el desarrollo de una serie de manuales y otra documentación para Linux, incluyendo las páginas de manual. Estos manuales se encuentran en diversos estados de desarrollo, y cualquier ayuda de revisión y puesta al dia de los mismos es apreciada enormemente. Si tiene preguntas acerca del LDP, por favor contacte con Matt Welsh(`mdw@sunsite.unc.edu`).

Estos libros están disponibles a través de FTP anónimo desde un montón de servidores de archivos Linux, incluyendo `sunsite.unc.edu` en el directorio `/pub/Linux/docs/LDP`. Unos cuantos distribuidores comerciales están vendiendo copias impresas de estos libros; en el futuro, Ud. podrá encontrar los manuales del LDP en las estanterías de su librería más próxima.

Linux Installation and Getting Started, de Matt Welsh
: Una nueva guía de usuario para Linux, que cubre todo lo que un nuevo usuario necesita saber para comenzar. Este libro lo tiene ahora mismo en sus manos.

The Linux System Administrators' Guide, de Lars Wirzenius
: Esta es una guía completa para poner en marcha y configurar un sistema Linux. Hay muchos temas relacionados con la administración de sistemas que son específicas de Linux, tales como lo necesario para soportar una comunidad de usuarios, mantenimiento del sistema de ficheros, backups, y demás. Esta guía los cubre a todos.

The Linux Network Administrators' Guide, de Olaf Kirch
: Una extensa y completa guía de redes bajo Linux, que incluye TCP/IP, UUCP, SLIP, y más. Este libro es de lectura recomendable; contiene gran cantidad de información sobre muchos temas, y clarifica la gran cantidad de aspectos confusos de la configuración en red.

The Linux Kernel Hackers' Guide, de Michael Johnson
: Hasta los más pequeños detalles del núcleo por dentro y del desarrollo bajo Linux. Linux es único en poner a disposición los fuentes del núcleo al completo. Este libro abre las puertas a desarrolladores que desean añadir o modificar características dentro del núcleo. Esta guía también da una amplia covertura a los conceptos del núcleo y las convenciones usadas en Linux.

A.3 Libros y Otros Trabajos Publicados

Linux Journal es una revista mensual para y sobre la comunidad Linux, escrito y producido por algunos de los desarrolladores y entusiastas de Linux. Se distribuye por el mundo entero, y es una forma excelente de mantenerse en contacto con la dinámica del mundo de Linux, especialmente si no se tiene acceso a las news de USENET.

En la fecha que se escribe esto, las subscripciones a *Linux Journal* son de US$22/año en los Estados Unidos, US$27 en Canada, y US$32 en el resto. Para subscribirse, o para ampliar información, escriba a Linux Journal, PO Box 85867, Seattle, WA, 98145-1867, USA, o llame al +1 206 527-3385. El número de FAX es +1 206 527-2806, y la dirección e-mail es `linux@ssc.com`. También podrá encontrar un *Linux Journal* FAQ y algunos artículos de muestra vía FTP anónimo en `sunsite.unc.edu` en `/pub/Linux/docs/linux-journal`.

Como ya hemos dicho, no se han publicado muchos libros que tengan que ver con Linux concretamente. Sin embargo, si Ud. es nuevo en el mundo de UNIX, o desea más información de la que se presenta aquí, le sugerimos que eche un vistazo a los siguientes libros que podrá encontrar.

A.3.1 Usando UNIX

Título: *Learning the UNIX Operating System*
Autor: Grace Todino & John Strang
Editorial: O'Reilly and Associates, 1987
ISBN: 0-937175-16-1, $9.00

Un buen libro de introducción al aprendizaje del sistema operativo UNIX. La mayoría de la información se debería de poder aplicar también a Linux. Le sugeriría leer este libro si Ud. es nuevo en UNIX y realmente quiere iniciarse en el uso de este nuevo sistema.

Título: *Learning the* `vi` *Editor*
Autor: Linda Lamb
Editorial: O'Reilly and Associates, 1990
ISBN: 0-937175-67-6, $21.95

Este es un libro sobre el editor `vi`, un potente editor de texto que podrá encontrar en todos los sistemas UNIX del mundo. A menudo es importante saber y poder usar `vi`, porque no siempre tendrá acceso a un editor "real" como el Emacs.

A.3.2 Administración de Sistemas

Título: *Essential System Administration*
Autor: Æleen Frisch
Editorial: O'Reilly and Associates, 1991
ISBN: 0-937175-80-3, $29.95

From the O'Reilly and Associates Catalog, "Como cualquier otro sistema multiusuario, UNIX requiere algo de cuidado y dedicación. *Essential System Administration* te dice cómo. Este libro destierra el mito y la confusión que rodea a este importante tópico y proporciona una introducción compacta y manejable de las tareas a que se enfrenta cualquier responsable de un sistema UNIX." Ni yo mismo podría haberlo dicho mejor.

Título: *TCP/IP Network Administration*
Autor: Craig Hunt
Editorial: O'Reilly and Associates, 1990
ISBN: 0-937175-82-X, $24.95

Una guía completa para poner a punto y echar a andar una red TCP/IP. A pesar de que este libro no es concretamente de Linux, cerca del 90% de él es aplicable a Linux. Junto con el Linux NET-2-HOWTO y el *Linux Network Administrator's Guide*, este es un gran libro que trata los conceptos y los detalles técnicos del manejo de TCP/IP.

Título: *Managing UUCP and Usenet*
Autor: Tim O'Reilly and Grace Todino
Editorial: O'Reilly and Associates, 1991
ISBN: 0-937175-93-5, $24.95

Este libro cubre como instalar y configurar el software de red UUCP, incluyendo la configuración de las news de USENET. Si está absolutamente interesado en usar UUCP o acceder a las news de USENET en su sistema, debería leer este libro.

A.3.3 The X Window System

Título: *The X Window System: A User's Guide*
Autor: Niall Mansfield
Editorial: Addison-Wesley
ISBN: 0-201-51341-2, ??

Un tutorial completo y una guía de referencia para usar el Sistema X Window. Si ha instalado X window en su sistema Linux, y quiere saber como sacar el máximo de él, debería leer este libro. Al contrario que algunos sistemas de ventanas, mucha de la potencia que proporciona X no es evidente a primera vista.

A.3.4 Programación

Título: *The C Programming Language*
Autor: Brian Kernighan and Dennis Ritchie
Editorial: Prentice-Hall, 1988
ISBN: 0-13-110362-8, $25.00

Este libro es obligado para cualquiera que desee programar en C en un sistema UNIX. (O cualquier sistema, pensandolo mejor.) Aunque este libro no es claramente específico de UNIX, es perfectamente aplicable a la programación de C bajo UNIX.

Título: *The Unix Programming Environment*
Autor: Brian Kernighan and Bob Pike

Editorial: Prentice-Hall, 1984
ISBN: 0-13-937681-X, ??

Una panorámica de la programación bajo sistemas UNIX. Da covertura a todas las herramientas disponibles; una buena lectura para familiarizarse con el a veces amorfo mundo de la programación de UNIX.

Título: *Advanced Programming in the UNIX Environment*
Autor: W. Richard Stevens
Editorial: Addison-Wesley
ISBN: 0-201-56317-7, $50.00

Este importante tomo contiene todo lo que necesita saber para programar UNIX a nivel del sistema—E/S de ficheros, control de procesos, comunicación entre procesos, señales, E/S del terminal, los trabajos. Este libro se enfoca sobre varios estandards UNIX, incluyendo POSIX.1, al cual se adhiere Linux en la mayoría de los casos.

A.3.5 Manejo del Kernel

Título: *The Design of the UNIX Operating System*
Autor: Maurice J. Bach
Editorial: Prentice-Hall, 1986
ISBN: 0-13-201799-7, ??

Este libro cubre los algoritmos e interioridades del kernel de UNIX. No es específico de ningún kernel en particular, aunque se inclina hacia el System V. Es el mejor punto de comienzo para entender las triquiñuelas del sistema Linux.

Título: *The Magic Garden Explained*
Autor: Berny Goodheart and James Cox
Editorial: Prentice-Hall, 1994
ISBN: 0-13-098138-9, ??

Este libro describe el kernel del Sistem V R4 al detalle. Al contrario que el libro de Bach, que se concentra profundamente en los algoritmos que dan vida al kernel, este libro presenta la implementación del SVR4 en un nivel más técnico. Aunque Linux y SVR4 son primos lejanos, este libro puede dar una visión muy profunda de los trabajos de una implementación actual de un kernel de UNIX. Este es también un libro muy moderno sobre el kernel de UNIX—publicado en 1994.

Apéndice B

Lista de Vendedores de Linux

Este apéndice lista información de contacto de un número de vendedores que ofrecen Linux en disquete, cinta, y CD-ROM. Muchos de ellos proporcionan documentación de Linux, soporte, así como otros servicios. En ningún modo ésta es una lista completa; si Ud. compró este libro impreso, es posible que la compañía de ventas o publicaciones también le haya proporcionado el software y servicios de Linux.

El autor no garantiza la exactitud de cualquier información listada en este Apéndice. Esta información se incluye aquí únicamente como un servicio a los lectores, no como un anuncio de una organización concreta.

Fintronic Linux Systems
1360 Willow Rd., Suite 205
Menlo Park, CA 94025 USA
Tel: +1 415 325-4474
Fax: +1 415 325-4908
`linux@fintronic.com`

InfoMagic, Inc.
PO Box 30370
Flagstaff, AZ 86003-0370 USA
Tel: +1 800 800-6613, +1 602 526-9565
Fax: +1 602 526-9573
`Orders@InfoMagic.com`

Lasermoon Ltd
2a Beaconsfield Road, Fareham,
Hants, England. PO16 0QB.
Tel: +44 (0) 329 826444.
Fax: +44 (0) 329 825936.
`info@lasermoon.co.uk`

Linux Journal
P.O. Box 85867
Seattle, WA 98145-1867 USA
Tel: +1 206 527-3385
Fax: +1 206 527-2806
`linux@ssc.com`

Linux Systems Labs
18300 Tara Drive
Clinton Twp, MI 48036 USA
Tel: +1 313 954-2829, +1 800 432-0556
Fax: +1 313 954-2806
`info@lsl.com`

Morse Telecommunication, Inc.
26 East Park Avenue, Suite 240
Long Beach, NY 11561 USA
Tel: +1 800 60-MORSE
Fax: +1 516 889-8665
`Linux@morse.net`

Nascent Technology
Linux from Nascent CDROM
P.O. Box 60669
Sunnyvale CA 94088-0669 USA
Tel: +1 408 737-9500
Fax: +1 408 241-9390
`nascent@netcom.com`

Red Hat Software
P.O. Box 4325
Chapel Hill, NC 27515 USA
Tel: +1 919 309-9560
`redhat@redhat.com`

SW Technology
251 West Renner Suite 229
Richardson, TX 75080 USA
Tel: +1 214 907-0871
`swt@netcom.com`

Takelap Systems Ltd.
The Reddings, Court Robin Lane,
Llangwm, Usk, Gwent, United Kingdom NP5 1ET.

Tel: +44 (0)291 650357
Fax: +44 (0)291 650500
info@ddrive.demon.co.uk

Trans-Ameritech Enterprises, Inc.
2342A Walsh Ave
Santa Clara, CA 95051 USA
Tel: +1 408 727-3883
roman@trans-ameritech.com

Unifix Software GmbH
Postfach 4918
D-38039 Braunschweig
Germany
Tel: +49 (0)531 515161
Fax: +49 (0)531 515162

Yggdrasil Computing, Incorporated
4880 Stevens Creek Blvd., Suite 205
San Jose, CA 95129-1034 USA
Tel: +1 800 261-6630, +1 408 261-6630
Fax: +1 408 261-6631
info@yggdrasil.com

Apéndice C

Tutorial de FTP y Lista de Sites

FTP ("File Transfer Protocol") es el conjunto de programas que se usa en Internet para transferir ficheros entre sistemas. La mayoría de los sistemas UNIX, VMS y MS-DOS de Internet tienen un programa llamado `ftp` que se usa para transferir estos ficheros, y si usted tiene acceso a Internet, el mejor modo de descargarse el software de Linux es usando `ftp`. Este apéndice cubre el uso de `ftp` a nivel básico—por supuesto, hay muchas más funciones y usos del `ftp` de los que se dan aquí.

Al final de este apéndice hay una lista de sites FTP donde se puede encontrar el software de Linux. Además, si no tiene acceso directo a internet, pero puede intercambiar correo electrónico con Internet, abajo se incluye información sobre como usar el `ftpmail`.

Si está usando un sistema MS-DOS, UNIX, o VMS para traer ficheros desde Internet, entonces `ftp` será un programa a base de comandos. Sin embargo, existen otras implementaciones de `ftp`, como la versión Macintosh (llamada `Fetch`) con un bonito interface guiado por menús, que lo hacen más amigable y cómodo de usar. Incluso si no esta usando la versión de comandos de `ftp`, la información que se facilita debería serle de ayuda.

`ftp` puede usarse tanto para "subir" (enviar) como para "bajar" (recibir) ficheros desde los sites de Internet. En la mayoría de situaciones, usted estará bajando software. En Internet hay un gran número de **FTP archive sites** de acceso público, máquinas que permiten a cualquiera hacer `ftp` sobre ellas y bajarse el software. Un "archive site" de este tipo es `sunsite.unc.edu`, que contiene un montón de software de Sun Microsistems, y que actúa como uno de los principales sites de Linux. Además, los FTP archive sites se reflejan (**mirror**) el software unos a otros—es decir, el software que se sube a un site será automáticamente copiado a un gran número de otros sites. De modo que no se sorprenda si ve exactamente los mismos ficheros en muchos sites diferentes.

C.1 Aprendiendo `ftp`

Fijese que en el ejemplo "screens" que figura debajo solamente estoy mostrando la información más relevante, y lo que usted vea podría diferir. Tome nota también de que los comandos en *itálicas* representan comandos que usted teclea; todo lo demás es la salida de pantalla.

Para arrancar el `ftp` y conectar con un site, use simplemente el comando

`ftp` ⟨*hostname*⟩

donde ⟨*hostname*⟩ es el nombre del site al que se quiere conectar. Por ejemplo, para conectar con el ya mítico site `shoop.vpizza.com` se puede usar el comando

```
ftp shoop.vpizza.com
```

C.2 Registrandose

Cuando comienza el `ftp` debería verse algo como

```
Connected to shoop.vpizza.com.
220 Shoop.vpizza.com FTPD ready at 15 Dec 1992 08:20:42 EDT
Name (shoop.vpizza.com:mdw):
```

En este punto, `ftp` nos está pidiendo que le demos el nombre de usuario con el que nos queremos registrar en `shoop.vpizza.com`. El nombre por defecto aquí es `mdw`, que corresponde con mi nombre de usuario del sistema desde el que estoy usando el FTP. Como no tengo cuenta en `shoop.vpizza.com` no me podré registrar como yo mismo. En cambio, para acceder al software disponible publicamente en un FTP site hay que registrarse como `anonymous`, y dar la dirección de e-mail Internet (si se tiene) como password. De modo que teclearíamos

```
Name (shoop.vpizza.com:mdw):  anonymous
331-Guest login ok, send e-mail address as password.
Password:  mdw@sunsite.unc.edu
230- Welcome to shoop.vpizza.com.
230- Virtual Pizza Delivery[tm]:  Download pizza in 30 cycles or less
230- or you get it FREE!
ftp>
```

Por supuesto, usted debería dar su dirección de e-mail en vez de la mia, y no se reflejará en pantalla mientras la esté escribiendo (ya que técnicamente es un "password" o contraseña). El `ftp` nos debería permitir registrarnos y ya estariamos preparados a bajarnos el software.

C.3 Moviendonos dentro

Perfecto, ya estamos dentro. Nuestro prompt es `ftp>`, y el programa `ftp` está a la espera de comandos. Hay unos pocos comandos básicos que debería conocer. Primero, los comandos

`ls` ⟨*fichero*⟩

y

`dir` ⟨*fichero*⟩

ambos dan un listado de ficheros (donde ⟨*fichero*⟩ es un argumento opcional que especifica un fichero particular a listar). La diferencia es que `ls` generalmente produce un listado corto y `dir` produce un listado más largo (es decir, con más información sobre los tamaños de los ficheros, fechas de modificación, etc.).

El comando

`cd` ⟨*directorio*⟩

moverá al directorio dado (exactamente como el comando `cd` en sistemas UNIX o MS-DOS). Puede usar también el comando

`cdup`

para cambiar al directorio padre[1].

El comando

`help` ⟨*comando*⟩

proporciona ayuda en el citado ⟨*comando*⟩ `ftp` (como puede ser `ls` o `cd`). Si no se especifica comando, `ftp` listará todos los comandos disponibles.

Si tecleamos `dir` en este punto, veremos un listado del directorio inicial en el que nos encontramos.

```
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 1337


dr-xr-xr-x  2 root     wheel         512 Aug 13 13:55 bin
drwxr-xr-x  2 root     wheel         512 Aug 13 13:58 dev
drwxr-xr-x  2 root     wheel         512 Jan 25 17:35 etc
drwxr-xr-x 19 root     wheel        1024 Jan 27 21:39 pub
drwxrwx-wx  4 root     ftp-admi     1024 Feb  6 22:10 uploads
drwxr-xr-x  3 root     wheel         512 Mar 11  1992 usr

226 Transfer complete.
921 bytes received in 0.24 seconds (3.7 Kbytes/s)
ftp>
```

[1] el directorio por encima del actual.

Cada una de estas entradas es un directorio, no un fichero individual que nos podamos traer (especificado por la `d` de la primera columna del listado). En la mayoría de los "FTP archive sites" (Servidores de FTP), el software públicamente accesible está bajo el directorio `/pub`, de modo que vamos a él.

```
ftp> cd pub
ftp> dir
200 PORT command successful.
150 ASCII data connection for /bin/ls (128.84.181.1,4525) (0 bytes).
total 846


-rw-r--r--   1 root     staff        1433 Jul 12  1988 README
-r--r--r--   1 3807     staff       15586 May 13  1991 US-DOMAIN.TXT.2
-rw-r--r--   1 539      staff       52664 Feb 20  1991 altenergy.avail
-r--r--r--   1 65534    65534       56456 Dec 17  1990 ataxx.tar.Z
-rw-r--r--   1 root     other     2013041 Jul  3  1991 gesyps.tar.Z
-rw-r--r--   1 432      staff       41831 Jan 30  1989 gnexe.arc
-rw-rw-rw-   1 615      staff       50315 Apr 16  1992 linpack.tar.Z
-r--r--r--   1 root     wheel       12168 Dec 25  1990 localtime.o
-rw-r--r--   1 root     staff        7035 Aug 27  1986 manualslist.tblms
drwxr-xr-x   2 2195     staff         512 Mar 10 00:48 mdw
-rw-r--r--   1 root     staff        5593 Jul 19  1988 t.out.h

226 ASCII Transfer complete.
2443 bytes received in 0.35 seconds (6.8 Kbytes/s)
ftp>
```

Aquí podemos ver un número de (¿interesantes?) ficheros, uno de los cuales se llama `README`, que deberíamos bajarnos (la mayoría de FTP sites tienen un fichero `README` en el directorio `/pub`).

C.4 Traerse ficheros

Antes de traer ficheros, hay algunas cosas que debería tener en cuenta.

- **Conectar las marcas de progreso.** Las *Marcas de Progreso* se imprimen en pantalla mientras se están transfiriendo ficheros; le permitirán saber como de avanzada va su transferencia, y que su conexión no se ha cortado (de modo que no esté sentado 20 minutos pensando que aun está trayendose un fichero). En general, una marca de progreso aparece como una almohadilla (`#`), y se imprime una por cada 1024 o 8192 bytes transferidos, dependiendo del sistema.

  Para activar la impresión de marcas de progreso, utilice el comando `hash`.

  ```
  ftp> hash
  Hash mark printing on (8192 bytes/hash mark).
  ftp>
  ```

- **Determinar el tipo de fichero que va a transferir.** En lo que concierne al FTP, los ficheros van en dos formatos: *binario* y *texto*. La mayoría de los ficheros que usted transferirá serán binarios: es decir, programas, ficheros comprimidos, ficheros de datos, etc. Sin embargo, muchos ficheros (tales como `README`s y demás) son ficheros de texto.

  ¿Por qué es importante el tipo de fichero? Simplemente porque en algunos sistemas (como el MS-DOS), ciertos caracteres de un fichero de texto, tales como retornos de carro, necesitan convertirse de modo que los ficheros puedan ser legibles. Mientras las transferencias en modo binario, no realizan ninguna conversión—el fichero se tranfiere byte a byte.

  Los comandos `bin` y `ascii` ponen el modo de tranferencia en binario y texto respectivamente. *Si está en duda, use siempre el modo binario para transferir ficheros.* Si intenta tranferir un fichero binario en modo texto, obtendrá un fichero corrupto que no podrá utilizar. (Este es uno de los fallos más corrientes cuando se usa el FTP.) Sin embargo, puede usar tranquilamente el modo texto para ficheros de texto (aquellos cuyos nombres terminan habitualmente en `.txt`).

  En nuestro ejemplo traeremos el fichero `README`, que por lo general es un fichero de texto, para lo cual usaremos el comando

  ```
  ftp> ascii
  200 Type set to A.
  ftp>
  ```

- **Establecer el directorio local.** El *directorio local* es el directorio de su sistema en el que quiere que vayan a parar los ficheros traidos. Mientras el comando `cd` cambia el directorio remoto (de la máquina remota a la que está haciendo FTP), el comando `lcd` cambia el directorio local.

  Por ejemplo, para cambiar el directorio local a `/home/db/mdw/tmp`, utilice el siguiente comando

  ```
  ftp> lcd /home/db/mdw/tmp
  Local directory now /home/db/mdw/tmp
  ftp>
  ```

Ahora ya está listo para traerse el fichero. El comando

`get` ⟨*nombre-remoto*⟩ ⟨*nombre-local*⟩

se usa para esto, donde ⟨*nombre-remoto*⟩ es el nombre del fichero de la máquina remota, y ⟨*nombre-local*⟩ es el nombre que le quiere dar al fichero en su máquina local. El argumento ⟨*nombre-local*⟩ es opcional; por defecto el nombre fichero local es el mismo que el remoto. Sin embargo, si está trayendose el fichero `README`, y ya tiene un `README` en su directorio local, necesitará darle un ⟨*nombre-local*⟩ distinto para que no se sobreescriba el primero.

En nuestro ejemplo, para traer el fichero `README`, usamos simplemente

```
ftp> get README
200 PORT command successful.
150 ASCII data connection for README (128.84.181.1,4527) (1433 bytes).
```

```
#
226 ASCII Transfer complete.
local:  README remote:  README
1493 bytes received in 0.03 seconds (49 Kbytes/s)
ftp>
```

C.5 Saliendo de FTP

Para terminar una sesión FTP, solo tiene que usar el comando

```
quit
```

El comando

```
close
```

se puede usar para cerrar la conexión con el FTP site actual; el comando `open` se puede usar para comenzar una sesión con otro site distinto (sin salir completamente del programa FTP).

```
ftp> close
221 Goodbye.
ftp> quit
```

C.6 Usando `ftpmail`

`ftpmail` es un servicio que le permite obtener ficheros de FTP sites a través del correo electrónico de Internet. Si no tiene acceso directo a Internet, pero al menos puede enviar correo a Internet (desde un servicio como CompuServe, por ejemplo), entonces `ftpmail` es un buen método para conseguir los ficheros de los FTP archive sites. Desafortunadamente, `ftpmail` puede ser lento, sobre todo cuando envía grandes trabajos. Antes de intentar traerse gran cantidad de software usando `ftpmail`, asegurese de que su cola de correo pueda manejar el tráfico entrante. Muchos sistemas ponen cuotas de correo entrante, y pueden borrar su cuenta si su correo se excede de esta cuota. Solamente tiene que usar el sentido común.

`sunsite.unc.edu`, uno de los más grandes FTP sites de Linux, posee un servidor de `ftpmail`. Para usar este servicio, envie un e-mail a

```
ftpmail@sunsite.unc.edu
```

en cuyo cuerpo del mensaje contenga solamente la palabra:

```
help
```

Esto le enviará de vuelta una lista de comandos `ftpmail` y un breve tutorial de uso del sistema.

Por ejemplo, para conseguir una lista de los ficheros de Linux encontrados en `sunsite.unc.edu`, envíe un correo a la dirección mencionada y que contenga el siguiente texto

```
open sunsite.unc.edu
cd /pub/Linux
dir
quit
```

Podrá usar el servicio `ftpmail` para conectar con cualquier FTP site; no está limitado a `sunsite.unc.edu`. La siguiente sección lista un número de FTP sites de Linux.

C.7 Lista de FTP Sites de Linux

La tabla C.1 es una lista de los más conocidos FTP sites que mantienen software de Linux. Tenga en cuenta también que un gran número de otros sites son copia de estos, y muy probablemente podrá encontrar Linux en un montón de sites que no están en esta lista.

Nombre del Site	Dirección IP	Directorio
tsx-11.mit.edu	18.172.1.2	/pub/linux
sunsite.unc.edu	152.2.22.81	/pub/Linux
nic.funet.fi	128.214.6.100	/pub/OS/Linux
ftp.mcc.ac.uk	130.88.200.7	/pub/linux
fgb1.fgb.mw.tu-muenchen.de	129.187.200.1	/pub/linux
ftp.informatik.tu-muenchen.de	131.159.0.110	/pub/Linux
ftp.dfv.rwth-aachen.de	137.226.4.105	/pub/linux
ftp.informatik.rwth-aachen.de	137.226.112.172	/pub/Linux
ftp.ibp.fr	132.227.60.2	/pub/linux
kirk.bu.oz.au	131.244.1.1	/pub/OS/Linux
ftp.uu.net	137.39.1.9	/systems/unix/linux
wuarchive.wustl.edu	128.252.135.4	/systems/linux
ftp.win.tue.nl	131.155.70.100	/pub/linux
ftp.ibr.cs.tu-bs.de	134.169.34.15	/pub/os/linux
ftp.denet.dk	129.142.6.74	/pub/OS/linux

Tabla C.1: Linux FTP Sites

`tsx-11.mit.edu`, `sunsite.unc.edu`, y `nic.funet.fi` son los “home sites” del software de Linux, donde se deja la mayoría del nuevo software. La mayoría de los otros sites de la lista se hacen copia de alguna combinación de estos tres. Para reducir el tráfico en la red, elija un site que esté geográficamente lo más cerca de usted.

Apéndice D

Lista de BBS de Linux

Aquí se ha puesto una lista de bulletin board systems (BBS) que mantienen software de Linux. Esta lista la mantiene Zane Healy (`healyzh@holonet.net`). Si conoce o tiene una BBS que proporcione software de Linux, y que no esté en esta lista, debería contactar con él.

La comunidad Linux ya no es solamente una sociedad únicamente de Internet. De hecho, se estima que la mayoría de usuarios de Linux no tienen acceso a internet. Por ello es especialmente importante que las BBSs continuen proporcionando y apoyando a los usuarios de Linux a lo ancho del mundo.

D.1 Estados Unidos

Citrus Grove Public Access, 916-381-5822. ZyXEL 16.8/14.4 Sacramento, CA. Internet: `citrus.sac.ca.us`
Higher Powered BBS, 408-737-7040. ? CA. RIME ->HIGHER
hip-hop, 408-773-0768. 19.2k Sunnyvale, CA. USENET access
hip-hop, 408-773-0768. 38.4k Sunnyvale, CA.
Unix Online, 707-765-4631. 9600 Petaluma, CA. USENET access
The Outer Rim, 805-252-6342. Santa Clarita, CA.
Programmer's Exchange, 818-444-3507. El Monte, CA. Fidonet
Programmer's Exchange, 818-579-9711. El Monte, CA.
Micro Oasis, 510-895-5985. 14.4k San Leandro, CA.
Test Engineering, 916-928-0504. Sacramento, CA.
Slut Club, 813-975-2603. USR/DS 16.8k HST/14.4K Tampa, FL. Fidonet 1:377/42
Lost City Atlantis, 904-727-9334. 14.4k Jacksonville, FL. FidoNet
Aquired Knowledge, 305-720-3669. 14.4k v.32bis Ft. Lauderdale, FL. Internet, UUCP
The Computer Mechanic, 813-544-9345. 14.4k v.32bis St. Petersburg, FL. Fidonet, Sailnet, MXBBSnet
AVSync, 404-320-6202. Atlanta, GA.
Information Overload, 404-471-1549. 19.2k ZyXEL Atlanta, GA. Fidonet 1:133/308

Atlanta Radio Club, 404-850-0546. 9600 Atlanta, GA.
Rebel BBS, 208-887-3937. 9600 Boise, ID.
Rocky Mountain HUB, 208-232-3405. 38.4k Pocatello, ID. Fionet, SLNet, CinemaNet
EchoMania, 618-233-1659. 14.4k HST Belleville, IL. Fidonet 1:2250/1, f'req LINUX
UNIX USER, 708-879-8633. 14.4k Batavia, IL. USENET, Internet mail
PBS BBS, 309-663-7675. 2400 Bloomington, IL.
Third World, 217-356-9512. 9600 v.32 IL.
Digital Underground, 812-941-9427. 14.4k v.32bis IN. USENET
The OA Southern Star, 504-885-5928. New Orleans, LA. Fidonet 1:396/1
Channel One, 617-354-8873. Boston, MA. RIME ->CHANNEL
VWIS Linux Support BBS, 508-793-1570. 9600 Worcester, MA.
WayStar BBS, 508-481-7147. 14.4k V.32bis USR/HST Marlborough, MA. Fidonet 1:333/14
WayStar BBS, 508-481-7293. 14.4k V.32bis USR/HST Marlborough, MA. Fidonet 1:333/15
WayStar BBS, 508-480-8371. 9600 V.32bis or 14.4k USR/HST Marlborough, MA. Fidonet 1:333/16
Programmer's Center, 301-596-1180. 9600 Columbia, MD. RIME
Brodmann's Place, 301-843-5732. 14.4k Waldorf, MD. RIME ->BRODMANN, Fidonet
Main Frame, 301-654-2554. 9600 Gaithersburg, MD. RIME ->MAINFRAME
1 Zero Cybernet BBS, 301-589-4064. MD.
WaterDeep BBS, 410-614-2190. 9600 v.32 Baltimore, MD.
Harbor Heights BBS, 207-663-0391. 14.4k Boothbay Harbor, ME.
Part-Time BBS, 612-544-5552. 14.4k v.32bis Plymouth, MN.
The Sole Survivor, 314-846-2702. 14.4k v.32bis St. Louis, MO. WWIVnet, WWIVlink, etc
MAC's Place, 919-891-1111. 16.8k, DS modem Dunn, NC. RIME ->MAC
Digital Designs, 919-423-4216. 14.4k, 2400 Hope Mills, NC.
Flite Line, 402-421-2434. Lincoln, NE. RIME ->FLITE, DS modem
Legend, 402-438-2433. Lincoln, NE. DS modem
MegaByte Mansion, 402-551-8681. 14.4 V,32bis Omaha, NE.
Mycroft QNX, 201-858-3429. 14.4k NJ.
Steve Leon's, 201-886-8041. 14.4k Cliffside Park, NJ.
Dwight-Englewood BBS, 201-569-3543. 9600 v.42 Englewood, NJ. USENET
The Mothership Cnection, 908-940-1012. 38.4k Franklin Park, NJ.
The Laboratory, 212-927-4980. 16.8k HST, 14.4k v.32bis NY. FidoNet 1:278/707
Valhalla, 516-321-6819. 14.4k HST v.32 Babylon, NY. Fidonet (1:107/255), UseNet (`die.linet.org`)
Intermittent Connection, 503-344-9838. 14.4k HST v.32bis Eugene, OR. 1:152/35
Horizon Systems, 216-899-1086. USR v.32 Westlake, OH.
Horizon Systems, 216-899-1293. 2400 Westlake, OH.
Centre Programmers Unit, 814-353-0566. 14.4k V.32bis/HST Bellefonte, PA.
Allentown Technical, 215-432-5699. 9600 v.32/v.42bis Allentown, PA. WWIVNet 2578
Tactical-Operations, 814-861-7637. 14.4k V32bis/V42bis State College, PA. Fidonet 1:129/226, `tac_ops.UUCP`
North Shore BBS, 713-251-9757. Houston, TX.
The Annex, 512-575-1188. 9600 HST TX. Fidonet 1:3802/217

The Annex, 512-575-0667. 2400 TX. Fidonet 1:3802/216
Walt Fairs, 713-947-9866. Houston, TX. FidoNet 1:106/18
CyberVille, 817-249-6261. 9600 TX. FidoNet 1:130/78
splat-ooh, 512-578-2720. 14.4k Victoria, TX.
splat-ooh, 512-578-5436. 14.4k Victoria, TX.
alaree, 512-575-5554. 14.4k Victoria, TX.
Ronin BBS, 214-938-2840. 14.4 HST/DS Waxahachie (Dallas), TX. RIME, Intelec, Smartnet, etc.
VTBBS, 703-231-7498. Blacksburg, VA.
MBT, 703-953-0640. Blacksburg, VA.
NOVA, 703-323-3321. 9600 Annandale, VA. Fidonet 1:109/305
Rem-Jem, 703-503-9410. 9600 Fairfax, VA.
Enlightend, 703-370-9528. 14.4k Alexandria, VA. Fidonet 1:109/615
My UnKnown BBS, 703-690-0669. 14.4k V.32bis VA. Fidonet 1:109/370
Georgia Peach BBS, 804-727-0399. 14.4k Newport News, VA.
Top Hat BBS, 206-244-9661. 14.4k WA. Fidonet 1:343/40
victrola.sea.wa.us, 206-838-7456. 19.2k Federal Way, WA. USENET

D.2 Fuera de los Estados Unidos

Galaktische Archive, 0043-2228303804. 16.8 ZYX Wien, Austria. Fidonet 2:310/77 (19:00-7:00)
Linux-Support-Oz, +61-2-418-8750. v.32bis 14.4k Sydney, NSW, Austrailia. Internet/Usenet, E-Mail/News
500cc Formula 1 BBS, +61-2-550-4317. V.32bis Sydney, NSW, Australia.
Magic BBS, 403-569-2882. 14.4k HST/Telebit/MNP Calgary, AB, Canada. Internet/Usenet
Logical Solutions, 299-9900 through 9911. 2400 AB, Canada.
Logical Solutions, 299-9912, 299-9913. 14.4k Canada.
Logical Solutions, 299-9914 through 9917. 16.8k v.32bis Canada.
V.A.L.I.S., 403-478-1281. 14.4k v.32bis Edmonton, AB, Canada. USENET
The Windsor Download, (519)-973-9330. v32bis 14.4 ON, Canada.
r-node, 416-249-5366. 2400 Toronto, ON, Canada. USENET
Synapse, 819-246-2344. 819-561-5268 Gatineau, QC, Canada. RIME->SYNAPSE
Radio Free Nyongwa, 514-524-0829. v.32bis ZyXEL Montreal, QC, Canada. USENET, Fidonet
DataComm1, +49.531.132-16. 14.4 HST Braunschweig, NDS, Germany. Fido 2:240/550, LinuxNet
DataComm2, +49.531.132-17. 14.4 HST Braunschweig, NDS, Germany. Fido 2:240/551, LinuxNet
Linux Server /Braukmann, +49.441.592-963. 16.8 ZYX Oldenburg, NDS, Germany. Fido 2:241/2012, LinuxNet
MM's Spielebox, +49.5323.3515. 14.4 ZYX Clausthal-Zfd., NDS, Germany. Fido 2:241/3420
MM's Spielebox, +49.5323.3516. 16.8 ZYX Clausthal-Zfd., NDS, Germany. Fido 2:241/3421
MM's Spielebox, +49.5323.3540. 9600 Clausthal-Zfd., NDS, Germany. Fido 2:241/3422
Bit-Company / J. Bartz, +49.5323.2539. 16.8 ZYX MO Clausthal-Zfd., NDS, Germany. Fido

2:241/3430
Fractal Zone BBS /Maass, +49.721.863-066. 16.8 ZYX Karlsruhe, BW, Germany. Fido 2:241/7462
Hipposoft /M. Junius, +49.241.875-090. 14.4 HST Aachen, NRW, Germany. Fido 2:242/6, 4:30-7,8-23:30
UB-HOFF /A. Hoffmann, +49.203.584-155. 19.2 ZYX+ Duisburg, Germany. Fido 2:242/37
FORMEL-Box, +49.4191.2846. 16.8 ZYX Kaltenkirchen, SHL, Germany. Fido 2:242/329, LinuxNet (6:00-20:00)
BOX/2, +49.89.601-96-77. 16.8 ZYX Muenchen, BAY, Germany. Fido 2:246/147, info magic: LINUX (22-24,0:30-2,5-8)
Die Box Passau 2+1, +49.851.555-96. 14.4 V32b Passau, BAY, Germany. Fido 2:246/200 (8:00-3:30)
Die Box Passau Line 1, +49.851.753-789. 16.8 ZYX Passau, BAY, Germany. Fido 2:246/2000 (8:00-3:30)
Die Box Passau Line 3, +49.851.732-73. 14.4 HST Passau, BAY, Germany. Fido 2:246/202 (5:00-3:30)
Die Box Passau ISDN, +49.851.950-464. 38.4/64k V.110/X.75 Passau, BAY, Germany. Fido 2:246/201 (8:00-24:00,1:00-3:30)
Public Domain Kiste, +49.30.686-62-50. 16.8 ZYX BLN, Germany. Fido 2:2403/17
CS-Port / C. Schmidt, +49.30.491-34-18. 19.2 Z19 Berlin, BLN, Germany. Fido 2:2403/13
BigBrother / R. Gmelch, +49.30.335-63-28. 16.8 Z16 Berlin, BLN, Germany. Fido 2:2403/36.4 (16-23:00)
CRYSTAL BBS, +49.7152.240-86. 14.4 HST Leonberg, BW, Germany. Fido 2:2407/3, LinuxNet
Echoblaster BBS #1, +49.7142.213-92. HST/V32b Bietigheim, BW, Germany. Fido 2:2407/4, LinuxNet (7-19,23-01h
Echoblaster BBS #2, +49.7142.212-35. V32b Bietigheim, BW, Germany. Fido 2:2407/40, LinuxNet (20h-6h)
LinuxServer / P. Berger, +49.711.756-275. 16.8 HST Stuttgart, BW, Germany. Fido 2:2407/34, LinuxNet (8:3-17:5,19-2)
Rising Sun BBS, +49.7147.3845. 16.8 ZYX Sachsenheim, BW, Germany. Fido 2:2407/41, LinuxNet (5:30-2:30)
bakunin.north.de, +49.421.870-532. 14.4 D 2800 Bremen, HB, Germany. `kraehe@bakunin.north.de`
oytix.north.de, +49.421.396-57-62. ZYX HB, Germany. `mike@oytix.north.de`, login as `gast`
Fiffis Inn BBS, +49-89-5701353. 14.4-19.2 Munich, Germany. FidoNet 2:246/69,Internet,USENET,LinuxNet
The Field of Inverse Chaos, +358 0 506 1836. 14.4k v32bis/HST Helsinki, Finland. USENET; ichaos.nullnet.fi
Modula BBS, +33-1 4043 0124. HST 14.4 v.32bis Paris, France.
Modula BBS, +33-1 4530 1248. HST 14.4 V.32bis Paris, France.
STDIN BBS, +33-72375139. v.32bis Lyon, Laurent Cas, France. FidoNet 2:323/8
Le Lien, +33-72089879. HST 14.4/V32bis Lyon, Pascal Valette, France. FidoNet 2:323/5
Basil, +33-1-44670844. v.32bis Paris, Laurent Chemla, France.
Cafard Naum, +33-51701632. v.32bis Nantes, Yann Dupont, France.

DUBBS, +353-1-6789000. 19.2 ZyXEL Dublin, Ireland. Fidonet 2:263/167

Galway Online, +353-91-27454. 14.4k v32b Galway, Ireland. RIME, `@iol.ie`

Nemesis' Dungeon, +353-1-324755 or 326900. 14.4k v32bis Dublin, Ireland. Fidonet 2:263/150

nonsolosoftware, +39 51 6140772. v.32bis, v.42bis Italy. Fidonet 2:332/407

nonsolosoftware, +39 51 432904. ZyXEL 19.2k Italy. Fidonet 2:332/417

Advanced Systems, +64-9-379-3365. ZyXEL 16.8k Auckland, New Zealand. Singet, INTLnet, Fidonet

Thunderball Cave, 472567018. Norway. RIME ->CAVE

DownTown BBS Lelystad, +31-3200-48852. 14.4k Lelystad, Netherlands. Fido 2:512/155, UUCP

MUGNET Intl-Cistron BBS, +31-1720-42580. 38.4k Alphen a/d Rijn, Netherlands. UUCP

The Controversy, (65)560-6040. 14.4k V.32bis/HST Singapore. Fidonet 6:600/201

Pats System, +27-12-333-2049. 14.4k v.32bis/HST Pretoria, South Africa. Fidonet 5:71-1/36

Gunship BBS, +46-31-693306. 14.4k HST DS Gothenburg Sweden.

Baboon BBS, +41-62-511726. 19.2k Switzerland. Fido 2:301/580 and /581

The Purple Tentacle, +44-734-590990. HST/V32bis Reading, UK. Fidonet 2:252/305

A6 BBS, +44-582-460273. 14.4k Herts, UK. Fidonet 2:440/111

On the Beach, +444-273-600996. 14.4k/16.8k Brighton, UK. Fidonet 2:441/122

Apéndice E

Linux en España

Este apéndice contiene varias fuentes de información tales como libros, BBS, forums en castellano, etc. La recopilación está basada en el documento "Fuentes de Información Linux en España" que periodicamente publica Ramón Gutiérrez Camus [1] en el "echo" español R34.LINUX. Diríjase a él si desea una información más actualizada.

E.1 Internet Sites en España

Suelen ser mirrors de los lugares "oficiales". La Tabla E.1 lista los FTP españoles que disponen de información y/o ficheros sobre Linux.

Nombre	Dirección IP	Directorio
luna.gui.uva.es.		
ftp.upc.es		
ocaso.sis.ucm.es		
asterix.fi.upm.es	138.100.8.6)	/pub/linux
rigel.deusto.es	130.206.100.2	
diable.upc.es		/pub/sistemes/Linux
sunsite.rediris.es		

Tabla E.1: Linux FTP Sites en España

E.2 BBS Españoles con ficheros Linux

Relación de BBS españoles que disponen de ficheros Linux en mayor o menor grado.

LuckyLink BBS Barcelona (93) 4400899, ¡BBS montada en Linux!

[1] Puede contactar con Ramón Gutiérrez en rcamus@jet.es ó 2:344/3.439@fidonet.org

Deusto BBS Bilbao - Tlf (94) 4130397, Dispone de ficheros Linux directamente recibidos de Internet por alumnos de esta Universidad. No admite FileRequest.

Europa BBS 2:341/19 - Madrid (91)320.84.60,

Virtual Software BBS, 2:341/8 - Madrid (91) 355-16-12 y (91) 355-28-95

New Order (2:345/207) Málaga (95) 238 13 43.

ArFIN BBs (2:341/68) Madrid (34-1) 326.2605, 28k8 VFC - 24 Horas. Acceso libre a ficheros Linux. No Acepta FileRequest. Dispone de ocho areas dedicadas con 52 Mb de ficheros Linux. Pedir FILES para lista de ficheros.

NorCom-Zombi BBS (2:348/103) La Coruña (???) ?????? Tiene un area de ficheros y, comentandoselo a su SysOp, permite el acceso a su sistema linux.

FIC BBS (94:348/1) La Coruña (???) ??????

Rias Baixas BBS, Pontevedra (2:348/603),+34-(9)86-863434, 24 horas. 28k8 bps V.34 V.FC V.32terbo H16 V.32bis V.42Bis

Telebase STD (2:345/102) Almería, Tlf (950) 274.830, 28k8 V34 - 24 Horas.

Area Soft BBS (2:340/21) Las Palmas (928) 201351

Apéndice F

The GNU General Public License

A continuación le mostramos la Licencia Publica General GNU (La `GPL`[1] o *copyleft* [2]), a la cual está sometido el Linux. Se reproduce aquí para aclarar algunas de las confusiones que se dan sobre el estado del copyright de Linux—Linux *no* es shareware, y *no* está en el dominio publico. El grueso del núcleo de Linux está bajo copyright ©1993 de Linus Torvalds, y otro software y partes del núcleo están bajo copyright de sus autores. En este caso, Linux *tiene* copyright, sin embargo, Ud. puede distribuirlo en los términos de la GPL que se imprime a continuación en su versión original.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

F.1 Preámbulo

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it,

[1] N. del T.: Del inglés General Public License

[2] N. del T.: Juego de palabras en inglés que quiere decir que es algo como un Copyright... pero al revés

[3] N. del T.: No tenemos conocimiento de la existencia de una versión oficial de la GPL en español, por lo que hemos dejado la licencia original, intacta y en inglés.

that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

F.2 Términos y Condiciones para la Copia, Distribución y Modificación

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

   In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly

through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

F.3 Apéndice: Cómo aplicar estos términos a sus nuevos programas

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> ⟨*one line to give the program's name and a brief idea of what it does.*⟩ Copyright ©19yy ⟨*name of author*⟩
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with
ABSOLUTELY NO WARRANTY; for details type ‘show w’.  This is free software, and
you are welcome to redistribute it under certain conditions; type ‘show c’ for
details.
```

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items–whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

> 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

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Glosario

bugs Errores en la confección de un programa.

driver programa encargado de servir de intermediario entre el sistema operativo y los distintos dispositivos conectados en el ordenador.

FAQ Frecuent Asked Question. Cuestión frecuentemente preguntada. Recopilación de preguntas y respuestas sobre las dudas más frecuentes.

FTP File Transfer Protocol: Protocolo utilizado para la transmisión de ficheros.

GNU GNU 's not UNIX. Frase recursiva que representa al proyecto encargado de la protección y creación de software de libre distribución.

hacker En su doble acepción, persona que posee un nivel muy importante de conocimientos en Unix, experto. También se dice que un hacker es aquel que utiliza sus extensos conocimientos de informática para "romper" sistemas de seguridad.

HOWTO Como ir hacia... como conseguir... Documentacion específica sobre algún aspecto concreto de Linux (impresoras, redes, UPS, etc.

kernel núcleo. En Linux, parte principal del sistema operativo. Código fuente del propio sistema

LDP Linux Documentation Project: Proyecto para la Documentación de Linux.

LILO LInux LOader. Cargador de Linux. Pequeño programa que permite el arranque de uno o varios sistemas operativos situados en una misma máquina. LILO puede poner en marcha, OS/2, Linux, ms-dos, Windows'95, SCO, etc.

LSM Linux Software Map. "Mapa" de software para Linux. Base de datos que recoge la mayor parte del software disponible para Linux, así como su autor, descripción, localización, observaciones, etc.

mainframe Ordenador principal. Equipo de muy altas prestaciones, Host.

núcleo kernel.

NFS Network File System: Sistemas de ficheros en red.

online on-line: en línea, en conexión. Documentación on-line: documentos disponibles a través de sistemas telemáticos

script Guión. Shell script: guión para el procesador de órdenes. Programa batch. Equivalente en unix a los ficheros BAT de MS-DOS.

shell Intérprete de órdenes o comandos. Programa intermediario entre el sistema operativo y el usuario. Imagínatelo como si fuera el sustituto del COMMAND.COM de ms-dos. Linux soporta varias shells diferentes (bash, ksh, cshell...).

stdin entrada estándar; usualmente el teclado.

stdout salida estándar; usualmente la pantalla.

swap Intercambio. swap space: Area de intercambio. Memoria virtual. Zona de almacenamiento en disco (fichero o partición) que sirve como prolongación de la memoria RAM realmente instalada, permitiendo la ejecución de procesos que, de otra forma, requerirían una ampliación física de dicha RAM.

Índice de Materias