

*CURSO
INTRODUCCTORIO DE
LINUX*

Facilitador:



CURSO
INTRODUCTORIO DE
LINUX

Introducción

Un sistema operativo es la parte software encargada de **gestionar los recursos hardware del sistema** y proporciona la base sobre la que se pueden escribir programas de aplicación. Entre las funciones típicas de un Sistema Operativo, se pueden destacar:

- ⊙ Gestión de la memoria
- ⊙ Control de las operaciones de Entrada y Salida
- ⊙ Tratamiento de Interrupciones
- ⊙ Control del Sistema
- ⊙ Establecimiento de prioridades y ejecución de procesos

Como ejemplos de sistemas operativos tenemos DOS, OS2, UNIX, MVS, WINDOWS NT, LINUX, MAC OS, etc.

El hardware solo se puede controlar a través de ceros y unos. Esto resulta complicado de manejar para cualquier usuario. Para simplificar estas tareas los computadores equipan en memoria ROM (solo lectura) una serie de instrucciones, de 50 a 300, en ensamblador (jump, move, add, etc.) que se traducen directamente a código binario.

Los sistemas operativos se pueden construir a partir de instrucciones en ensamblador o a partir de lenguajes de más alto nivel (como C, Pascal, Basic, etc.). Los lenguajes de alto nivel no son entendidos por los computadores directamente, por lo que hay que compilarlos, es decir convertirlos a un formato que si pueda ser entendido.

Una de las funciones principales de un sistema operativo es ocultar toda la complejidad que existe por debajo y proporcionar al programador un conjunto de instrucciones fáciles de usar.

Los comandos se pueden definir como programas que acompañan al sistema operativo sin formar parte de él, y que están diseñados para trabajar con los demás comandos o con el kernel (o núcleo) del Sistema Operativo.

Por encima del sistema operativo se encuentran aplicaciones y programas, que utilizan los servicios proporcionados por el sistema operativo. Como ejemplos de aplicaciones nos encontramos con Windows, PowerPoint, Staroffice, cualquier juego, etc.

Los computadores los podemos clasificar según su potencia de menor a mayor en:

- ⊙ Microcomputadores: computadores personales, PC, etc
- ⊙ Minicomputadores: Computadores multiusuario, VAX, 3B2, etc.
- ⊙ Mainframes: Computadores con mayor potencia de procesamiento, IBM 370, etc.
- ⊙ Supercomputadores: Computadores de gran potencia utilizados en aplicaciones científicas, CRAY.

Historia de UNIX

A pesar de los sistemas abiertos, la historia de UNIX está dominada por el ascenso y caída de los sistemas hardware. UNIX nació en 1969 en una mainframe 635 de General Electric. A la vez, los Laboratorios Bell de AT&T había completado el desarrollo de Multics, un sistema multiusuario que falló por su gran demanda de disco y memoria. En respuesta a Multics, los ingenieros de sistemas Kenneth Thompson y Dennis Ritchie inventaron el UNIX. Inicialmente, Thompson y Ritchie diseñaron un sistema de archivos para su uso exclusivo, pero pronto lo cargaron en una Digital Equipment Corp. (DEC) PDP-7, una computadora con solo 18 kilobytes de memoria. Este suministraba una larga serie de puertos. En 1970, fue cargado en una PDP-11, y el runoff, el predecesor del troff, se convirtió en el primer procesador de texto de UNIX. En 1971, UNIX recibió reconocimiento oficial de AT&T cuando la firma lo usó para escribir manuales.

La segunda edición de UNIX fue realizada en 1971. La segunda edición dio forma al UNIX moderno con la introducción del lenguaje de programación C y sobre los 18 meses siguientes, el concepto de los pipes. Los pipes fueron importantes por muchas razones. Representaron una nueva forma de tratamiento de datos. Desde un punto de vista moderno, los pipes son un mecanismo orientado a objetos, porque entregan datos desde un objeto, o programa, a otro objeto.

Mientras tanto, había mucha actividad con el C. El C es otro producto de los Laboratorios Bell. Fue formado a partir de conceptos de otros tres lenguajes: B, CPL (Combined Programming Language) y Algol-60. A finales de 1973, después de que Ritchie añadió soporte para variables globales y estructuras, C se convirtió en el lenguaje de programación de UNIX preferente. (Brian Kernigham, quien ayudó a Ritchie a desarrollar el C, añadió la R al estándar K&R, es el estándar preferido hasta la aceptación del ANSI C).

¿Qué es Linux?

LINUX es un sistema operativo, compatible Unix. Dos características muy peculiares lo diferencian del resto de los sistemas que podemos encontrar en el mercado, la primera, es que es *libre*, esto significa que no tenemos que pagar ningún tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo, la segunda, es que el sistema viene acompañado del *código fuente*. El sistema lo forman el Núcleo (Kernel) del sistema mas un gran número de programas / librerías que hacen posible su utilización.

Linux se distribuye bajo la GNU General Public License, por lo tanto, el código fuente tiene que estar siempre accesible.

El sistema ha sido diseñado y programado por multitud de programadores alrededor del mundo. El núcleo del sistema sigue en continuo desarrollo bajo la coordinación de Linus Torvalds, la persona de la que partió la idea de este proyecto, a principios de la década de los noventa.

Día a día, más y más programas / aplicaciones estan disponibles para este sistema, y la calidad de los mismos aumenta de versión a versión. La gran mayoría de los mismos vienen acompañados del código fuente y se distribuyen gratuitamente bajo los términos de licencia de la GNU Public License, aunque también existen programas comerciales, los cuales hay que comprar y que no se distribuyen bajo la licencia GNU.

En los últimos tiempos, ciertas casas de software comercial han empezado a distribuir sus productos para Linux y la presencia del mismo en empresas aumenta rapidamente por la excelente relación calidad-precio que se consigue con Linux.

Las plataformas en las que en un principio se puede utilizar Linux son 386-, 486-, Pentium, Pentium Pro, Pentium II/III, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como Alpha, ARM, MIPS, PowerPC y SPARC

Linux es una variante de UNIX, pero debe quedar claro de que Linux no es UNIX. Linux ha sido y sigue siendo desarrollado, por miles de programadores voluntarios alrededor del mundo, aunque también existen programadores pagados por las distribuciones.

La idea original, detrás de Linux, se remonta a principio de 1990, en la Universidad de Tecnología de Helsinki, Finlandia, por un estudiante Sueco, de nombre Linus Torvalds, que comenzó en 1991 como un proyecto para proveer de una alternativa, al sistema operativo Minix, que es un UNIX reducido.

En pocos años, el equipo de desarrolladores de Linux, se expandió increíblemente, ya no solo se desarrollaba drivers/kernel, sino también se creaban aplicaciones para este sistema operativo de código abierto, trasladando primeramente los programas de UNIX y luego creando nuevas aplicaciones.

Este explosivo crecimiento, hizo de la instalación y mantenimiento de Linux, una tarea verdaderamente difícil, lo que obligó a grupos de personas a reunir los kernel y software de Linux, para crear lo que hoy conocemos como Distribuciones Linux, que no es otra cosa que un grupo de software predefinidos y empaçados, con el sistema operativo Linux y con una utilidad de instalación, que lo hacía cada vez más accesible, al usuario final.

Sobre todo Linux viene a ser un amalgamador de un sentimiento común de libertad, vista en Internet, donde siempre existe alguien presto a ayudarnos con su experiencia, en la solución de los problemas que se nos presentan, esto es, "La Comunidad Linux".

Historia de Linux

LINUX hace su aparición a principios de la década de los noventa, era el año 1991, por aquel entonces un estudiante de informática de la Universidad de Helsinki, llamado Linus Torvalds empezó, -como una afición y sin poderse imaginar a lo que llegaría este proyecto, a programar las primeras líneas de código de este sistema operativo llamado LINUX.

Este comienzo estuvo inspirado en MINIX, un pequeño sistema Unix desarrollado por Andy Tanenbaum. Las primeras discusiones sobre Linux fueron en el grupo de noticias comp.os.minix, en estas discusiones se hablaba sobre todo del desarrollo de un pequeño sistema Unix para usuarios de Minix que querían más.

Linus nunca anunció la versión 0.01 de Linux (agosto 1991), esta versión no era ni siquiera ejecutable, solamente incluía los principios del núcleo del sistema, estaba escrita en lenguaje ensamblador y asumía que uno tenía acceso a un sistema Minix para su compilación.

El 5 de octubre de 1991, Linus anunció la primera versión "Oficial" de Linux, -versión 0.02. Con esta versión Linus pudo ejecutar Bash (GNU Bourne Again Shell) y gcc (El compilador GNU de C) pero no mucho más funcionaba. En este estado de desarrollo ni se pensaba en los términos soporte, documentación, distribución y desarrollo.

Después de la versión 0.03, Linus saltó en la numeración hasta la 0.10, más y más programadores a lo largo y ancho de internet empezaron a trabajar en el proyecto y después de sucesivas revisiones, Linus incrementó el número de versión hasta la 0.95 (Marzo 1992). Más de un año después (diciembre 1993) el núcleo del sistema estaba en la versión 0.99 y la versión 1.0 no llegó hasta el 14 de marzo de 1994.

Desde entonces no se ha parado de desarrollar, la versión actual del núcleo es la 2.2 y sigue avanzando día a día con la meta de perfeccionar y mejorar el sistema.

Características de Linux

- ⑩ **Multitarea:** La palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo.
Linux utiliza la llamada multitarea preeventiva, la cual asegura que todos los programas que se están utilizando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada programa.
- ⑩ **Multiusuario:** Muchos usuarios usando la misma máquina al mismo tiempo.
- ⑩ **Multiplataforma:** Las plataformas en las que en un principio se puede utilizar Linux son 386-, 486-, Pentium, Pentium Pro, Pentium II/III, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como Alpha, ARM, MIPS, PowerPC y SPARC.
- ⑩ **Multiprocesador:** Soporte para sistemas con más de un procesador está disponible para Intel y SPARC.
- ⑩ **Funciona en modo protegido 386.**

- ⑩ Protección de la memoria entre procesos, de manera que uno de ellos no pueda colgar el sistema.
- ⑩ Carga de ejecutables por demanda: Linux sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente.
- ⑩ Política de copia en escritura para la compartición de páginas entre ejecutables: esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página (4Kb de memoria) se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.
- ⑩ Memoria virtual usando paginación (sin intercambio de procesos completos) a disco: A una partición o un archivo en el sistema de archivos, o ambos, con la posibilidad de añadir más áreas de intercambio sobre la marcha. Un total de 16 zonas de intercambio de 128Mb de tamaño máximo pueden ser usadas en un momento dado con un límite teórico de 2Gb para intercambio. Este límite se puede aumentar fácilmente con el cambio de unas cuantas líneas en el código fuente.
- ⑩ La memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y ésta puede a su vez ser reducida cuando se ejecuten grandes programas.
- ⑩ Librerías compartidas de carga dinámica (DLL's) y librerías estáticas.
- ⑩ Se realizan volcados de estado (core dumps) para posibilitar los análisis post-mortem, permitiendo el uso de depuradores sobre los programas no sólo en ejecución sino también tras abortar éstos por cualquier motivo.
- ⑩ Compatible con POSIX, System V y BSD a nivel fuente.
- ⑩ Emulación de iBCS2, casi completamente compatible con SCO, SVR3 y SVR4 a nivel binario.
- ⑩ Todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito.
- ⑩ Control de tareas POSIX.
- ⑩ Pseudo-terminales (pty's).
- ⑩ Emulación de 387 en el núcleo, de tal forma que los programas no tengan que hacer

- ⑩ Control de tareas POSIX.
- ⑩ Pseudo-terminales (pty's).
- ⑩ Emulación de 387 en el núcleo, de tal forma que los programas no tengan que hacer su propia emulación matemática. Cualquier máquina que ejecute Linux parecerá dotada de coprocesador matemático. Por supuesto, si el computador ya tiene una FPU (unidad de coma flotante), esta será usada en lugar de la emulación, pudiendo incluso compilar tu propio kernel sin la emulación matemática y conseguir un pequeño ahorro de memoria.
- ⑩ Soporte para muchos teclados nacionales o adaptados y es bastante fácil añadir nuevos dinámicamente.
- ⑩ Consolas virtuales múltiples: varias sesiones de login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (totalmente independiente del hardware de video). Se crean dinámicamente y puedes tener hasta 64.
- ⑩ Soporte para varios sistemas de archivo comunes, incluyendo minix-1, Xenix y todos los sistemas de archivo típicos de System V, y tiene un avanzado sistema de archivos propio con una capacidad de hasta 4 Tb y nombres de archivos de hasta 255 caracteres de longitud.
- ⑩ Acceso transparente a particiones MS-DOS (o a particiones OS/2 FAT) mediante un sistema de archivos especial: no es necesario ningún comando especial para usar la partición MS-DOS, esta parece un sistema de archivos normal de Unix (excepto por algunas restricciones en los nombres de archivo, permisos, y esas cosas). Las particiones comprimidas de MS-DOS 6 no son accesibles en este momento, y no se espera que lo sean en el futuro. El soporte para VFAT, FAT32 (WNT, Windows 95/98) se encuentra soportado desde la versión 2.0 del núcleo y el NTFS de WNT desde la versión 2.2 (Este último solo en modo lectura).
- ⑩ Un sistema de archivos especial llamado UMSDOS que permite que Linux sea instalado en un sistema de archivos DOS.
- ⑩ Soporte en sólo lectura de HPFS-2 del OS/2 2.1
- ⑩ Sistema de archivos de CD-ROM que lee todos los formatos estándar de CD-ROM.
- ⑩ TCP/IP, incluyendo ftp, telnet, NFS, etc.
- ⑩ Appletalk.
- ⑩ Software cliente y servidor Netware.

Linux Copyright

El copyright del kernel de linux pertenece a Linus Torvalds. Este se distribuye bajo la GNU General Public License, esto significa básicamente, que cualquiera puede libremente, copiarlo, cambiarlo y distribuirlo, pero sin posibilidad de aplicar restricciones en futuras distribuciones y teniendo que incluir el código fuente.

Las licencias de los programas y aplicaciones que acompañan al núcleo varían, muchos pertenecen al proyecto GNU y se distribuyen bajo GPL, y otros no.

¿Cómo consigo Linux?

Linux es un sistema de libre distribución por lo que puedes encontrar todos los archivos/programas necesarios para su funcionamiento en multitud de servidores conectados a Internet. La tarea de reunir todos los archivos/programas necesarios, así como instalarlos en tu sistema puede ser una tarea bastante complicada y no apta para muchos. Por esto mismo, nacieron las llamadas distribuciones de Linux, empresas que se dedican a hacer el trabajo "sucio" para nuestro beneficio y comodidad.

Una distribución no es otra cosa, que una recopilación de programas y archivos, organizados y preparados para su instalación. Estas distribuciones se pueden obtener a través de Internet, o comprando los CDs de las mismas, los cuales contendrán todo lo necesario para instalar un sistema Linux bastante completo y en la mayoría de los casos un programa de instalación que nos ayudara en la tarea de una primera instalación. Casi todos los principales distribuidores de Linux, ofrecen la posibilidad de bajarse sus distribuciones, via FTP (sin cargo alguno).

Existen varias distribuciones creadas por diferentes empresas a unos precios bastantes asequibles (si se compran los CDs, en vez de bajársela via FTP), las cuales deberas de poder encontrar en tiendas de informática o librerías.

Principales distribuciones de linux:

©Redhat

Página web: <http://www.redhat.com>

FTP principal: <ftp://ftp.redhat.com/pub/>

©Debian

Página web: <http://www.debian.org/>

FTP principal: <ftp://ftp.debian.org/debian/>

©SuSE

Página web: <http://www.suse.de/es/>

FTP principal: <ftp://ftp.suse.com/>

FTP principal: <ftp://ftp.debian.org/debian/>

Ⓢ **SuSE**

Pagina web: <http://www.suse.de/es/>

FTP principal: <ftp://ftp.suse.com/>

Ⓢ **Caldera Openlinux**

Pagina web: <http://www.caldera.com/>

FTP principal: <ftp://ftp.caldera.com/pub/OpenLinux/>

Ⓢ **Slackware**

Pagina web: <http://www.slackware.com/>

FTP principal: <ftp://ftp.cdrom.com/pub/linux/slackware>

Ⓢ **Mklinux**

Pagina web: <http://www.mklinux.apple.com/>

FTP principal: <ftp://ftp.mklinux.apple.com/pub/>

Ⓢ **Mandrake (español)**

Pagina web: <http://www.linux-mandrake.com/es/>

FTP principal: <ftp://sunsite.uio.no/pub/unix/Linux/Mandrake/>

Ⓢ **Hispafuentes (español)**

Pagina web: <http://www.hispafuentes.com/home.php3>

Ⓢ **Conectiva (español)**

Pagina web: <http://www.conectiva.com/>

FTP principal: <ftp://ftp.conectiva.com/pub/conectiva/>

Ⓢ **Eurielec (español)**

Pagina web: <http://www.eurielec.etsit.upm.es/linux/>

FTP principal: <ftp://ftp.dit.upm.es/linux/>

Ⓢ **Esware (español)**

Pagina web: <http://www.esware.com/>

FTP principal: <ftp://ftp.hardlogic.es/pub/>

¿Funcionan mis programas/juegos para Windows en Linux?

Como primera afirmación, podemos decir que no, tus programas para Windows no funcionarían en Linux. Windows y Linux/Unix no son compatibles y programas compilados en una u otra plataforma no funcionarían en otra plataforma que no sea en la que el programa se compiló para su utilización.

Como segunda afirmación, podemos decir, que en algunos casos es posible ejecutar

¿Cuántas particiones necesito para Linux?

La respuesta rápida y fácil es: recomendable al menos dos, una para el sistema/datos y otra para Swap. Usualmente se suelen tener tres, una para el sistema/programas (/), otra para los datos (/home) y otra para swap.

¿Qué es la Swap?

La swap es un espacio reservado en tu disco duro para poder usarse como una extensión de memoria virtual de tu sistema. Es una técnica utilizada desde hace mucho tiempo, para hacer creer a los programas que existe más memoria RAM de la que en realidad existe. Es el propio sistema operativo el que se encarga de pasar datos a la swap cuando necesita más espacio libre en la RAM y viceversa.

¿Cómo utilizar LoadLin para el arranque de Linux?

A veces ciertos programas o "sistemas operativos" están diseñados para modificar el *Master Boot Record* (MBR) del computador sin tener en cuenta los contenidos actuales del mismo, lo que suele redundar en la eliminación de LILO y la imposibilidad del usuario de entrar en Linux. Para solucionar estos problemas, o por si simplemente deseamos entrar en Linux desde msdos, está disponible el programa LOADLIN.

Loadlin es un programa de MSDOS que podemos encontrar en el directorio dosutils de los CDs de Linux (en la propia distribución). Este programa permite arrancar Linux desde MSDOS a partir de él mismo, un archivo kernel de Linux (como vmlinuz, bzImage o zImage) y una partición Linux. Mediante él podemos hacernos un disco de seguridad para arrancar Linux cuando LILO sea borrado por otros S.O.:

Creamos un directorio a:\loadlin y copiamos allí el ejecutable Loadlin.exe. Copiamos también cualquier kernel/núcleo de linux que tengamos disponible (por ejemplo \vmlinuz o \boot\bzImage) al mismo diskette.

Cuando necesitemos arrancar desde el diskette, arrancamos en msdos y ejecutamos lo siguiente:

Formato de uso:

```
loadlin kernel root=particion
```

En nuestro caso:

```
a:  
loadlin bzImage root=/dev/hda4
```

```
loadlin kernel root=particion
```

En nuestro caso:

```
a:  
loadlin bzImage root=/dev/hda4
```

(cambiando cada cosa por su nombre correcto de archivo de kernel y partición donde tengamos instalado Linux). Esto lo podemos escribir en el mismo diskette en un archivo linux.bat, por ejemplo. Tras ejecutarlo, Linux arrancará normalmente con dicho kernel/núcleo y con los datos de la partición indicada, lo cual permite restaurar el MBR original (con LILO) ejecutando de nuevo LILO (mediante /sbin/lilo, o /sbin/lilo -v).

¿Qué es LILO y para qué sirve?

LILO, literalmente, *el cargador de linux* (Linux LOader) es un gestor de arranque muy versátil que permite iniciar un sistema operativo (SO) cargando el sector de arranque de una partición del disco duro (o de un disquete).

¿Cómo se configura e instala LILO?

La configuración de LILO se hace en /etc/lilo.conf, pero antes que editar y escribir manualmente la configuración en este archivo, existen interfaces o programas capaces de preparar una configuración conveniente para el usuario, de manera que solamente en el caso de incorporar opciones avanzadas tenga que editar y modificar a mano el archivo.

En primer lugar existe un programa llamado "Quickinst" que acompaña la distribución original del programa LILO. Se trata de un guión (shell script) que produce una configuración mínima de LILO que luego conviene actualizar. Por otra parte la mayor parte de las distribuciones proporcionan un asistente durante la instalación que ayuda a la configuración de LILO. RedHat en su "INSTALL" dedica una parte importante a la configuración de LILO. El "SETUP" de Slackware también lo permite. Distribuciones como Caldera OpenLinux o SuSE disponen de herramientas de instalación que permiten una cómoda configuración de LILO para el usuario (LISA y LIZARD para OpenLinux 1.3 y 2.2, y YAST para SuSE). Quizá Debian ha sido un poco más espartana (al menos hasta Hamm 2.0) en su utilidad de instalación para la configuración de LILO: se limita a escribir una configuración básica de LILO sin soporte para otros SOs ni otros kernels.

¿Qué es el kernel/núcleo?

El kernel o núcleo de linux se podría definir como el corazón de este sistema operativo.

utilizan.

- ⊗ Es el encargado de que podamos acceder a los periféricos/elementos de nuestro computador de una manera cómoda.

Existen dos *versiones* del Linux kernel:

- ⊗ *Versión de producción*: La versión de producción, es la versión estable hasta el momento. Esta versión es el resultado final de las versiones de desarrollo o experimentales.

Cuando el equipo de desarrollo del núcleo experimental, decide que ha conseguido un núcleo estable y con la suficiente calidad, se lanza una nueva versión de producción o estable. Esta versión es la que se debería utilizar para un uso normal del sistema, ya que son las versiones consideradas más estables y libres de fallos en el momento de su lanzamiento.

- ⊗ *Versión de desarrollo*: Esta versión es experimental y es la que utilizan los desarrolladores para programar, comprobar y verificar nuevas características, correcciones, etc. Estos núcleos suelen ser inestables y no se deberían usar, a no ser que sepas lo que haces.

Cómo *interpretar los números de las versiones*: Las versiones del núcleo se numeran con 3 números, de la siguiente forma: XX.YY.ZZ

- ⊗ XX: Indica la serie principal del núcleo. Hasta el momento solo existen la 1 y 2. Este numero cambia cuando la manera de funcionamiento del núcleo ha sufrido un cambio muy importante.
- ⊗ YY: Indica si la versión es de desarrollo o de producción. Un número impar, significa que es de desarrollo, uno par, que es de producción.
- ⊗ ZZ: Indica nuevas versiones dentro de una versión, en las que lo unico que se ha modificado, son fallos de programación /bugs.

Unos *ejemplos* nos ayudarán a entenderlo mejor:

- ⊗ ej 1: versión del núcleo 2.0.0: Núcleo de la serie 2 (XX=2), versión de producción 0 (YY=0 par), primera versión de 2.0 (ZZ=0)
- ⊗ ej 2: versión del núcleo 2.0.1: Núcleo de la serie 2, version 0, en el que se han corregido errores de programación presentes en la versión 2.0.0 (ZZ=1)
- ⊗ ej 3: version del nucleo 2.1.100: version 100 del nucleo de desarrollo 2.1.

¿Dónde consigo el kernel/núcleo?

El núcleo se puede bajar de un gran número de servidores en internet. El servidor principal es <http://www.kernel.org/>

¿Dónde consigo el kernel/núcleo?

El núcleo se puede bajar de un gran número de servidores en internet. El servidor principal es <http://www.kernel.org/>

¿Cómo se configura e instala el kernel/núcleo?

1. Bajarse la última versión.

NOTA: Si vas a instalar un núcleo de la serie 2.2.x, tienes que tener en cuenta que algunas distribuciones no están/estaban preparadas para hacer uso de esta serie. Si tu distribución no es de las que vienen preparadas, tienes que actualizar una serie de paquetes/programas antes de instalar el nuevo núcleo (mas información en la documentación que acompaña al núcleo). Las últimas distribuciones vienen todas preparadas para los núcleos de la serie 2.2.x y posteriores, esta nota solo es para los que tengan una distribución antigua.

2. Tener claro lo que vamos a hacer, leerse el documento HOWTO sobre el núcleo.
3. Tener claro las opciones que tenemos que configurar, para poder utilizar el hardware de nuestro sistema, así como las características que queremos utilizar. Por ejemplo, si no utilizamos un dispositivo SCSI, no tenemos que configurar nada en el apartado SCSI de nuestro núcleo. Así nos ahorramos espacio y tiempo.
4. Entrar como root: `su root`
5. Ir al directorio `/usr/src/`: `cd /usr/src/`
6. Copiar el archivo que bajastes, al directorio `/usr/src/`: `cp linux-xx.yy.zz.tar.gz .`
7. Descomprimirlo y desempaquetar: `tar -xvzpf linux-xx.yy.zz.tar.gz`

NOTA: El archivo `linux-xx.yy.zz.tar.gz` se desempaquetará en el directorio `/usr/src/linux`. Si ya existe un directorio llamado `/usr/src/linux` en tu sistema, renombrarlo, p.ej: `mv linux linux-old`. En algunas distribuciones, `usr/src/linux` es un enlace simbólico a `linux-x.y.z`, borrar este enlace simbólico. Es importante que no exista ningún directorio/enlace simbólico llamado `linux`, antes de desempaquetar la nueva versión. Si te has bajado el kernel comprimido con `bzip2`, tendras que descomprimirlo con `bunzip2 linux-xx.yy.zz.tar.bz2` antes de desempaquetarlo con `tar -xvf linux-xx.yy.zz.tar`

8. Entrar en `/usr/src/linux`: `cd /usr/src/linux`

Si tienes XWindow instalado, te recomiendo el último comando, si no, el segundo. Es recomendable que las opciones que vienen por defecto no sean tocadas si no se sabe lo que se hace, puedes pulsar Help en cada opción para obtener una descripción de la misma. Configurar las opciones que quieras tener en tu nuevo núcleo. Una vez terminada la configuración, grabar los cambios y salir del programa de configuración.

10. Una vez terminado el proceso de configuración, tenemos que compilar nuestro nuevo núcleo. Para ello hay que hacer lo siguiente:

```
make dep
make clean
make bzImage
```

11. Si en el proceso de configuración, elegimos alguna opción como módulo, tendremos que compilar/installar dichos módulos:

```
make modules
make modules_install
```

NOTA: No olvidar ejecutar como root el comando `depmod -a` la primera vez que arranques con el nuevo núcleo, para computar las dependencias entre módulos.

12. Ya tenemos el núcleo y los módulos compilados, ahora tenemos que instalarlo. Casi todo el mundo utiliza LILO para arrancar el sistema.

Todavía estamos en `/usr/src/linux`, ejecutar el comando `make install`, esto copiará el núcleo que acabamos de crear, a el directorio `/boot` de nuestro sistema, con el nombre `vmlinuz`, o como un enlace simbólico `vmlinuz -> vmlinuz-xx.yy.zz`

13. Ahora tenemos que configurar LILO para que reconozca el nuevo núcleo. Tendremos que modificar el archivo `/etc/lilo.conf`. Aquí tienes un ejemplo, del archivo `/etc/lilo.conf` antes de modificarlo:

```
boot=/dev/hda
prompt
timeout=50
image=/boot/vmlinuz-2.0.34
    label=linux
    root=/dev/hda1
    read-only
```

Y aquí como quedaría después de la modificación, para que reconozca nuestro nuevo núcleo al arrancar:

```
label=linux
root=/dev/hda1
read-only
```

Y aquí como quedaría después de la modificación, para que reconozca nuestro nuevo núcleo al arrancar:

```
boot=/dev/hda
prompt
timeout=50
image=/boot/vmlinuz
    label=nuevokernel
    root=/dev/hda1
    read-only
image=/boot/vmlinuz-2.0.34
    label=linux
    root=/dev/hda1
    read-only
```

14. Ahora solo tenemos que ejecutar el comando `/sbin/lilo` y arrancar el sistema de nuevo. Si tuviesemos algún problema con el nuevo núcleo, siempre podríamos arrancar con el antiguo escribiendo `linux` y pulsando ENTER cuando arrancamos y nos sale en pantalla `lilo`: De esta manera podemos entrar y ver que es lo que ha fallado.

NOTA: Recordar que existen multitud de opciones para configurar LILO, y que los ejemplos anteriores, son ejemplos. Puede que tu sistema necesite diferentes parámetros y opciones.

¿Qué es XWindow?

El protocolo XWindow fue desarrollado a mediados de los años 80 como respuesta a la necesidad de un interfaz gráfico transparente sobre todo para sistemas Unix. XWindow es el encargado de visualizar la información de manera gráfica y es totalmente independiente del sistema operativo (los sistemas Unix/Linux no necesitan de XWindow para funcionar, pudiendo trabajar en modo texto). La gran diferencia entre XWindow y la interfaz gráfica de otros sistemas operativos es que XWindow distribuye el procesamiento de aplicaciones, especificando un enlace cliente-servidor. El cliente X especificará "Que hacer" al servidor X, que se encargará de "Como hacerlo".

La gran ventaja de XWindow es que el servidor X de una aplicación y el cliente X donde trabajamos no tienen que estar en la misma máquina. Podemos estar utilizando XWindow en nuestra máquina, conectarnos a otra remota, ejecutar un programa en esta máquina remota y utilizar/ver este programa en nuestra máquina local. Todo esto independientemente de la plataforma/sistema operativo que el computador remoto utilice.

Cliente X

|

¿Cómo configuro XWindow?

Para utilizar el sistema en modo gráfico es necesario instalar XWindow en tu sistema. Existen servidores comerciales y bajo licencia GNU. En estos apartados nos basaremos en XWindow bajo licencia GNU, XFree86, ya que es el que se incluye en todas las distribuciones de Linux.

Una vez instalado XWindow tendremos que configurarlo para que funcione en nuestro sistema. El archivo de configuración del XWindow se llama XF86Config y generalmente se encuentra en el directorio `/etc/X11/`.

Unos datos que tienes que saber antes de configurar XWindow son:

- Ⓣ Datos técnicos de la tarjeta gráfica que utilizas, o por lo menos el nombre de la tarjeta. El programa `/usr/X11R6/bin/SuperProbe` puede ayudar a obtener información sobre la tarjeta.
- Ⓣ Datos técnicos del monitor que tienes, resoluciones/frecuencias de refresco verticales/horizontales a las que puede trabajar. Este punto es importante para evitar problemas posteriores con el monitor. Estos datos deberían de estar explicados en la documentación que viene con el monitor.

Antes de lanzar XWindow mediante el comando `startx` debemos asegurarnos de la correcta configuración de los diferentes parámetros del mismo (teclado, ratón, tarjeta de vídeo, etc.). Hay diferentes métodos para hacer esto.

En primer lugar tenemos los programas de configuración semi-automática. Según la distribución de Linux que utilicemos tendremos a nuestra disposición una serie de programas que nos permitirán realizar la configuración de XWindow de una manera más o menos sencilla:

- Ⓣ `/usr/X11R6/bin/xf86config` - Todas las distribuciones: Programa en modo texto.
- Ⓣ `/usr/X11R6/bin/XF86Setup` - SuSe, Redhat, Slackware: Programa en modo gráfico, tiene que arrancarse en una consola texto.
- Ⓣ `/usr/X11R6/bin/Xconfigurator` Redhat: Programa semi-gráfico con menus.

Estos programas nos preguntarán cosas básicas sobre el teclado, el monitor, la videomemoria o la tarjeta de vídeo instalada (lo harán cuando no sean capaces de detectarlas por sí mismos). Debemos responder siempre la opción más acertada y en cuanto a monitores, nunca especificar características superiores de las que disponemos. Un caso muy habitual en tarjetas muy nuevas es no encontrar un soporte exacto para los últimos chipsets, problema que se suele corregir acudiendo al soporte de framebuffer del kernel o esperando a que quienes trabajan en XFree86 obtengan por parte de los fabricantes la información sobre cómo programar sus tarjetas.

En ciertos casos también se nos puede preguntar si deseamos que el sistema arranque directamente en XWindow o incluso testear los diferentes modos de vídeo para ver que si

Un caso muy habitual en tarjetas muy nuevas es no encontrar un soporte exacto para los últimos chipsets, problema que se suele corregir acudiendo al soporte de framebuffer del kernel o esperando a que quienes trabajan en XFree86 obtengan por parte de los fabricantes la información sobre cómo programar sus tarjetas.

En ciertos casos también se nos puede preguntar si deseamos que el sistema arranque directamente en XWindow o incluso testear los diferentes modos de vídeo para ver que si son correctamente soportados.

Además para que estos programas realicen una correcta configuración el sistema debe estar ya correctamente configurado en cuanto a modo texto se refiere. Por ejemplo, Xconfigurator no activará el teclado en castellano para XWindow si no hemos especificado que el idioma en que queremos usar nuestro Linux es este mismo, mediante la orden

```
export LANG=es_ES
```

bien en la línea de comandos, o en el archivo `/etc/profile`. Cuando Xconfigurator detecta esta variable de entorno ya sabe qué lenguaje utiliza el usuario y por tanto configurará XWindow para el correcto uso del teclado en este lenguaje. Otras variables indican el tipo de ratón, etc. y suelen ser correctamente especificadas por el programa de instalación. En el caso de XF86Setup la configuración permite una mayor participación por parte del usuario, ya que en lugar de realizar una configuración automática proporciona una lista de opciones para cada apartado, de modo que si conocemos nuestro hardware es la mejor manera de configurar el sistema. Además, a la hora de elegir resoluciones de vídeo, este programa permite especificar cual es la resolución por defecto deseada mientras que Xconfigurator deja marcada la menor de ellas como "por defecto" (o la primera que seleccionemos).

El programa de configuración `xf86config` es el más antiguo de todos, y realiza la configuración mediante preguntas que se responderán con teclado. Nos preguntará la tarjeta de vídeo, monitor, rango de frecuencias del monitor, videomemoria y resoluciones de trabajo deseadas, así como el mapa del teclado a utilizar (es 102 = España 102 teclas).

Como siempre, el método más eficaz de configurar XWindow es directamente editando los archivos de configuración. Y es que otro método de configuración básico es sencillamente editar el archivo `/etc/X11/XF86Config`, que es en realidad lo que hacen todos los programas comentados anteriormente. En este archivo la configuración está dividida en secciones y mediante cualquier editor de texto llano es posible cambiar cualquier parámetro de XWindow.

Este archivo se divide en varias categorías o secciones (todas ellas con más subopciones muy bien comentadas en dicho archivo de configuración):

● Sección Files: Indica los caminos hacia las fuentes o archivos de colores.

```

Protocol      "Standard"
LeftAlt       Meta
RightAlt      Meta
ScrollLock    Compose
RightCtl      Control
XkbKeycodes   "xfree86"
XkbTypes      "default"
XkbCompat     "default"
XkbSymbols    "us (pc101) "
XkbGeometry   "pc"
XkbRules      "xfree86"
XkbModel      "pc101"
XkbLayout     "es"

```

Ⓞ Sección Pointer: Configuración del ratón:

Para un ratón microsoft mouse o un genius:

```

Protocol      "Microsoft"
Device        "/dev/mouse"

```

Ⓞ Sección Monitor: Datos sobre el monitor (frecuencias).

Ⓞ Sección Device: Indica el servidor X a utilizar: Por ejemplo para una tarjeta con chipset Mach64:

```

Section "Device"
    Identifier "Mach64 GB"
    VendorName "ATI"
    BoardName  "ATI 3d Rage II"
    VideoRam   8192
EndSection

```

Ⓞ Sección Screen: Modos de vídeo seleccionados: Por ejemplo para activar 320x200, 640x480, 800x600 y 1024x768 tanto a 8 como a 16 bpp de color, siendo usado por defecto el modo 1024x768x16 (el primero que aparezca en la lista):

```

Section "Screen"
    Driver      "accel"
    Device      "Mach64 GB"
    Monitor     "My Monitor"
    DefaultColorDepth 16
    Subsection "Display"
        Depth      16
        Modes       "1024x768" "800x600" "640x480" "320x200"
        ViewPort    0 0
        Virtual     1024 768
    EndSubsection
    Subsection "Display"
        Depth      8
        Modes       "1024x768" "800x600" "640x480" "320x200"
        ViewPort    0 0
        Virtual     1024 768
    EndSubsection
EndSection

```

```

EndSubsection
Subsection "Display"
  Depth      8
  Modes      "1024x768" "800x600" "640x480" "320x200"
  ViewPort   0 0
  Virtual    1024 768
EndSubsection
EndSection

```

Una vez creado este archivo, entramos en XWindow mediante el comando startx.

Del entorno XWindow se puede salir en cualquier momento mediante la combinación de teclas Ctrl+Alt+Borrar, o bien seleccionando LOGOUT en el menú del gestor de ventanas que utilicemos.

Para volver a las consolas de texto bastará con utilizar la combinación de teclas Ctrl+Alt+F1, +F2, etc (hasta F6 en las instalaciones por defecto). Para volver al terminal gráfico se utiliza de F7 en adelante (Ctrl+Alt+F7, +F8, etc.).

Si durante la instalación (o en la línea apropiada del archivo XF86Config) seleccionamos varias resoluciones de vídeo, es posible cambiar entre ellas mediante las teclas Ctrl+Alt+'+' y Ctrl+Alt+'-' (los signos del teclado numérico).

Además, si por defecto marcamos 8bpp (o 16 bpp) en las opciones, nada nos impide lanzar posteriormente XWindow en la profundidad de color deseada:

```

[egarcia@localhost]# startx -- -bpp 8
[egarcia@localhost]# startx -- -bpp 16
etc.

```

También es posible establecer la profundidad de color por defecto mediante una línea DefaultColorDepth = 16 en el archivo de configuración XF86Config, sección Screen.

Por último, saber que podemos lanzar más de una sesión de XWindow simultáneamente (en las consolas accesibles desde Ctrl+Alt+F8, +F9, etc.), mediante:

```

[egarcia@localhost]# startx -- :1
[egarcia@localhost]# startx -- :2
etc.

```

¿Cómo cambio la resolución por defecto?

En el archivo /etc/X11/XF86Config en la sección screen correspondiente al servidor que se está utilizando actualmente, en la subsección Display correspondiente a la Depth en que estemos trabajando, buscamos la línea:

```

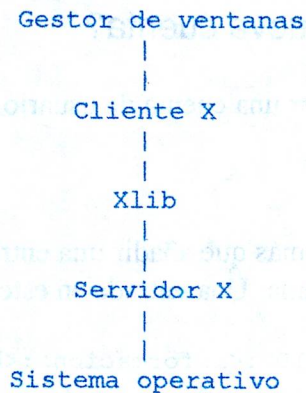
Modes "resolución_1" "resolución_2"...

```

¿Qué es un gestor de ventanas?

Un gestor de ventanas no es otra cosa que el conjunto de programas, ventanas, funcionalidades, que hacen posible que el usuario pueda interactuar con el sistema de forma gráfica y no en modo texto.

Para usar un gestor de ventanas, hay que tener configurado un *servidor X*. También hay que decir que el *gestor de ventanas* utilizado es totalmente independiente del *servidor X* utilizado.



Al contrario que en otros sistemas operativos, en Linux no es necesario utilizar un *servidor X - gestor de ventanas* para usar el sistema. El sistema operativo y el conjunto *servidor X - gestor de ventanas* usado, son cosas totalmente diferentes, independientes entre si. Es más, existen usuarios que trabajan en modo texto sin ningun problema y sin usar un interfaz gráfico.

Existen numerosos y variados gestores de ventanas para Linux, unos mejores y otros más desarrollados y estables. Es el usuario el que tiene que decidir que gestor satisface mejor sus necesidades, pudiendo incluso tener mas de uno instalado. Para aclarar un poco las cosas, podriamos decir que, si un computador es usado por varios usuarios, todos utilizarán el mismo *servidor X* pero no necesariamente el mismo *gestor de ventanas*.

Gestores de ventanas hay muchos: kde, afterstep, gnome, WindowMaker, icewm, etc, y todos tienen sus respectivos ejecutables: startkde, afterstep, gnome-session, wmaker, icewm, etc

¿Puedo trabajar normalmente como root?

Un consejo, solo utilizar la cuenta de root (administrador) para tareas de administración. Para trabajar normalmente con el sistema NO hacerlo como root, ya que se tiene acceso completo a todo el sistema y es fácil de estropear cosas si no se sabe lo que se hace. Lo mejor es trabajar como un usuario normal y cuando se necesite ser root hacer un `su`, y tan pronto como deje de ser necesario tener el privilegio hacer un `exit`, para volver a

¿Puedo trabajar normalmente como root?

Un consejo, solo utilizar la cuenta de root (administrador) para tareas de administración. Para trabajar normalmente con el sistema NO hacerlo como root, ya que se tiene acceso completo a todo el sistema y es fácil de estropear cosas si no se sabe lo que se hace. Lo mejor es trabajar como un usuario normal y cuando se necesite ser root hacer un su, y tan pronto como deje de ser necesario tener el privilegio hacer un exit, para volver a trabajar como usuario normal

¿Cómo abro/cierro una nueva cuenta?

Aquí veremos como abrir y cerrar una cuenta de usuario.

Abrir una nueva cuenta.

El abrir una nueva cuenta, no es más que añadir una entrada en el archivo/etc/passwd del sistema. Una entrada en este archivo tiene la siguiente forma:

```
<user>:<encrip>:<UID>:<GID>:<informacion>:<home>:<shell>
```

user:	Es el nombre de usuario.
encrip:	Es la clave encriptada.
UID:	Es el número UID con el que nos identificará el kernel.
GID:	El numero de grupo principal GID al que pertenecemos.
información:	Información varia, tal como nombre completo del usuario, número de teléfono, e-mail, etc.
home:	Es el directorio home del usuario, es decir, el directorio al que accedemos justo después de hacer entrar en el sistema.
shell:	Es el shell que se abrirá para ejecutar las órdenes que de el usuario.

Hay utilidades que hacen que añadir cuentas de usuario sea más fácil. Entre ellas están el comando /usr/sbin/adduser o /usr/sbin/useradd y /usr/bin/passwd. Lo pasos para crear un usuario son:

```
[root@asimovI /root] /usr/sbin/adduser mario  
[root@asimovI /root] passwd mario
```

Para más información hacer man adduser, sobre todo si se quiere poner caducidad a las claves.

Hay otras utilidades más vistosas y fáciles de usar e intuitivas ya que son bajo XWindow, un ejemplo de éstas son en el contol-panel de redhat, el YaST en SuSE, el gestor de usuarios de Kde, etc.

Borrar una cuenta de usuario.

Para borrar una cuenta de usuario también se pueden utilizar las utilidades gráficas que comentamos antes, de una forma muy intuitiva.

¿Cómo dar formato a un disquete en Linux?

El formato de disquetes en Linux depende del tipo de formato que se le desee dar al disquete (tipo MSDOS, tipo FAT32/VFAT, tipo Ext2 de Linux, minix, etc.). En general solo hay que usar uno de los comandos mkfs disponibles, que creará el sistema de archivos pertinente:

Para formatear disquetes en formato Linux-ext2

```
mkfs.ext2 [opciones] /dev/fd0
```

o bien su equivalente:

```
mke2fs [opciones] /dev/fd0
```

El usuario que formatee deberá tener permiso de escritura para la disquetera (normalmente /dev/fd0). Para eso debe estar en el grupo de usuarios floppy o que /dev/fd0 tenga permisos 666 (para todo el mundo).

Para formatear disquetes en formato msdos

```
mkfs.msdos [opciones] /dev/fd0
```

o bien:

```
mkdosfs [opciones] /dev/fd0
```

Otro comando disponible si tenemos instaladas las mtools es mformat:

```
mformat a:
```

Este último comando formateará el disquete en formato ms-dos.

Por último, existen otros programas como fdformat o superformat (ver paquete fdutils) que también sirven para dar formato a disquetes.

NOTA: Formatos del tipo ext2 y ms-dos son los que se utilizan más usualmente en linux. Hay que aclarar que un disquete con formato ext2 *No será posible leerlo* bajo ms-dos o Windows, solamente bajo Linux. Un disquete con formato ms-dos podrá ser leído indistintamente por ms-dos, windows o Linux (si tenemos soporte en el kernel/núcleo para trabajar con este formato).

fdutils) que también sirven para dar formato a disquetes.

NOTA: Formatos del tipo ext2 y ms-dos son los que se utilizan más usualmente en linux. Hay que aclarar que un disquete con formato ext2 *No será posible leerlo* bajo ms-dos o Windows, solamente bajo Linux. Un disquete con formato ms-dos podrá ser leído indistintamente por ms-dos, windows o Linux (si tenemos soporte en el kernel/núcleo para trabajar con este formato).

¿Cómo puedo montar y desmontar unidades en Linux?

Montar un sistema de archivos/dispositivo a nivel usuario no es más que hacerlo disponible en el árbol de directorios de nuestro sistema. Como ya sabe, en Linux vemos todos los sistemas de archivos/dispositivos en un sólo árbol de directorios, no existen letras a:, c:, etc., de esta manera nos da igual que el contenido de un directorio sea un sistema de archivos msdos, vfat, ext2, se encuentre en diferentes discos/particiones, esté en una máquina remota, etc. Esta abstracción tiene un inconveniente, hay que montarlo, es decir, indicarle al kernel de Linux que a través del directorio XXXX, accedemos al sistema de archivos/dispositivo yyy. Esto se hace con el comando mount (man mount, para detalles).

Montar un sistemas de archivos/dispositivo a nivel kernel, no es más que rellenar unas tablas de registro. Es decir, ver si el sistema de archivos está soportado, o lo que es lo mismo, si existe la tabla de funciones con las que manejarlo. Luego registrar estas funciones y enlazar el directorio al sistema de archivos. Esto es más bien lo que hace la llamada al sistema mount, la cual es llamada mediante el comando del mismo nombre.

Los parametros necesarios para montar un sistema de archivos son:

- ⑩ *Tipo de sistema de archivos:* Es necesario indicarle el tipo de sistemas de archivos a registrar. Así sabrá que funciones utilizar.
- ⑩ *Tipo de acceso:* Si es de sólo lectura, sólo escritura o ambos.
- ⑩ *Dispositivo:* Indicar el dispositivo sobre el que está físicamente el sistema de archivos,(p.e.: /dev/hda1, /dev/hdb, /dev/sda1,..., none si es el /proc, ya que se realiza sobre la memoria).
- ⑩ *Directorio:* Indica el directorio sobre el que se va a montar, es decir, el directorio mediante el cual accederemos al sistema de archivos.

Ejemplo: Montar el cdrom en el directorio /dev/cdrom.

```
mount -t <sistema de archivos> <Dispositivo> <Directorio>
mount -t iso9660 /dev/hdb /mnt/cdrom
```

Existen otras formas de reducir la línea de comandos, para ello (entre otras cosas) existe el archivo `/etc/fstab`. En él se indican los sistemas de archivos sobre los que trabajamos normalmente: el sistema de archivos en el que tenemos los directorios de linux, el `/proc`, la partición dos, el cdrom, y el floppy.

El archivo `/etc/fstab` funciona de la siguiente manera:

Partimos de un ejemplo de contenido de `/etc/fstab`:

```
# <device> <mountpoint> <filesystemtype> <options> <dump> <fsckorder>
/dev/hda2 / ext2 defaults 1 1
/dev/hda3 /usr ext2 defaults 1 2
/dev/sda1 /home ext2 defaults 1 2
/dev/hdb /mnt/cdrom iso9660 user,noexec,nodev,nosuid,ro,noauto 0 0
/dev/fd0 /mnt/floppy vfat user,noexec,nodev,nosuid,rw,noauto 0 0
none /proc proc defaults 0 0
/dev/hda4 swap swap defaults 0 0
/dev/hda1 /mnt/dos vfat exec,dev,suid,rw,auto 0 0
```

Con la información contenida en este archivo, el sistema haría lo siguiente al arrancar el sistema:

- ⊗ La partición `/dev/hda1` se montaría en el subdirectorio `/mnt/dos`
- ⊗ La partición `/dev/hda2` se montaría en el subdirectorio `/`
- ⊗ La partición `/dev/hda3` se montaría en el subdirectorio `/usr`
- ⊗ La partición `/dev/hda4` se montaría en el subdirectorio como `swap`
- ⊗ La partición `/dev/sda1` se montaría en el subdirectorio `/home`
- ⊗ Proc se montaría en el subdirectorio `/proc`
- ⊗ El sistema tendría información sobre como montar un disquete `/dev/fd0` y un CD-Rom `/dev/hdb`, aunque no los monta automáticamente al arrancar por haber definido la opción `noauto`.

Veamos la explicación de los parámetros usados en `/etc/fstab`:

En la columna de dispositivo (`device`) se indica el dispositivo/partición a montar, en la columna de montaje (`mountpoint`) se indica el directorio mediante el cual vamos a acceder al sistema de archivos. En la columna de tipo de sistema de archivos (`filesystemtype`) se indica el sistema de archivos que se usará sobre el dispositivo.

Las opciones (`options`) significan lo siguiente:

- ⊗ `user,nouser`: permite/no permite a un usuario ordinario montar el sistema de archivos.
- ⊗ `suid,nosuid`: Permite/no permite tener archivos con el bit de usuario definido.

indica el sistema de archivos que se usara sobre el dispositivo.

Las opciones (options) significan lo siguiente:

- ⓪ *user,nouser*: permite/no permite a un usuario ordinario montar el sistema de archivos.
- ⓪ *suid,nosuid*: Permite/no permite tener archivos con el bit de usuario definido.
- ⓪ *auto,noauto*: Indica que sí/no se monta cuando hacemos mount -a.
- ⓪ *defaults*: Aplica las opciones rw,suid,dev,exec,auto,nouser,async.
- ⓪ *exec,noexec*: Permite/no permite la ejecución de binarios.
- ⓪ *ro,rw*: Montar sólo lectura, lectura-escritura.
- ⓪ *sync/async*: Todos los accesos I/O al sistema de archivos se realizarán en modo sincrónico/asincrónico.
- ⓪ *dev/nodev*: Interpreta/no interpreta los dispositivos especiales de bloques/caracteres en el sistema de archivos.

El archivo `/etc/fstab` de nuestro ejemplo nos permitiría hacer lo siguiente para montar un disquete o CD-Rom:

```
mount /mnt/cdrom  
mount /mnt/floppy
```

Cuando dejamos de utilizar un sistema de archivos, tenemos que darlo de baja (desmontarlo), para eliminar todas las referencias del kernel a él. El comando utilizado para esto es `umount`.

Para desmontar un sistema de archivos/dispositivo:

```
umount <directorio|dispositivo>
```

ej:

```
umount /mnt/floppy  
umount /dev/fd0
```

NOTA: Es necesario ejecutar el comando `umount` cada vez que cambiemos un disquete, ya que no siempre cuando se escribe en un dispositivo, se realiza automáticamente un volcado al dispositivo (por eso se pueden perder datos si se apaga el computador a lo bruto). Además si cambiamos el disquete sin hacer `umount`, el kernel piensa que hay el sistema de archivos anterior y cuando guarda alguna información lo hace con referencia al retirado, perdiéndose todo.

Este es un punto a tener en cuenta con el uso de disquetes, ya que, el mecanismo para retirar un disquete es mecánico, no existe posibilidad de impedir por medio de software que alguien saque un disquete de la disquetera. No ocurre lo mismo con el CD-Rom, ya que el sistema bloqueará la posibilidad de extraerlo cuando este montado.

¿Cómo se utilizan los empaquetadores-des/compresores?

Los archivos tar no son archivos comprimidos, sino empaquetados. Tar es un empaquetador, es decir, es algo parecido a un compresor como arj o zip, pero sin compresión. Se dedica a incluir todos los archivos juntos en el mismo archivo, preservando las estructuras de directorios y permisos de los mismos. Como veremos, lo podremos comprimir gracias al programa GZip.

Hay 2 operaciones básicas con tar: empaquetado y desempaquetado. Si estamos en un directorio y queremos empaquetar todos los archivos de este directorio y los que cuelgan de él, basta con ejecutar la orden:

```
tar -cvf archivo.tar *

c = compress (más bien, empaquetar)
v = verbose (para que nos diga lo que hace)
f = file (empaquetar en un archivo)
* (empaquetar todos los archivos, podría haber sido
*.doc, etc.)
```

Si disponemos de un archivo .tar y queremos desempaquetarlo:

```
tar -xvf archivo.tar

x = eXtract (desempaquetar).
```

También es posible listar los contenidos de un archivo .tar antes de desempaquetarlo, mediante la orden `tar -tvf archivo.tar`.

Por otra parte, el archivos con extensión gz son archivos comprimidos. A diferencia de arj o zip, el contenido de un archivo GZ es un solo archivo, es decir, cuando comprimimos `archivo.txt` con este compresor (llamado gzip) obtenemos un `archivo.txt.gz` de tamaño mucho menor. Con GZ no es posible empaquetar archivos, es decir, la compresión se realiza a un sólo archivo.

Para comprimir un archivo con gz, se utiliza el comando:

```
gzip archivo
```

Para descomprimirlo:

```
gunzip archivo.gz
```

La combinación de tar y gz es lo que permite el tener multiples archivos comprimidos en un sólo archivo. Es decir, si empaquetamos un directorio con tar y luego comprimimos ese archivo tar con gz, obtenemos un `tar.gz` comprimido con múltiples archivos.

La compresión y descompresión es posible hacerla en 2 pasos (primero tar y luego usar

La combinación de tar y gz es lo que permite el tener multiples archivos comprimidos en un sólo archivo. Es decir, si empaquetamos un directorio con tar y luego comprimimos ese archivo tar con gz, obtenemos un tar.gz comprimido con múltiples archivos.

La compresión y descompresión es posible hacerla en 2 pasos (primero tar y luego usar gz) o bien usar el flag -z de tar para ello:

Compresión:

```
tar -cvzf archivo.tar.gz *
```

Descompresion:

```
tar -xvzf archivo.tar.gz
```

Otro formato que se ha puesto de moda es bzip2, con el mismo sistema de funcionamiento que Gzip, y cuyos nombres de ejecutable son bzip2 (comprimir) y bunzip2 (descomprimir). Este compresor obtiene mejor compresión que Gzip y su funcionamiento es igual de sencillo, aunque tarda mas en comprimir y utiliza mas recursos.

Estos compresores/descompresores/empaquetadores son una gran y libre alternativa a formatos comerciales como zip, arj y rar, también disponibles para Linux (comandos zip, unzip, rar y unarj).

Para descomprimir archivos arj mediante unarj, simplemente hace falta ejecutar el comando unarj x archivo.arj. El compresor es shareware y se debe obtener en la Web de sus programadores.

Zip es el programa destinado a hacer Linux capaz de leer y escribir los archivos en formato .zip (generados por pkzip o winzip): Para ello tenemos los comandos zip e unzip, que nos permitan comprimir y descomprimir archivos sueltos, directorios completos, directorios con recursividad, etc:

Para comprimir todos los archivos de un directorio en un zip:

```
zip archivo.zip *
```

Para comprimir este directorio y todos los que cuelguen del mismo:

```
zip -r archivo.zip *
```

La descompresión se realiza mediante unzip:

```
unzip archivo.zip
```

El programa rar también es un buen compresor que podemos encontrar en diferentes

Descomprimir:

```
rar x archivo
```

Para más información sobre cualquiera de los des/compresores basta con consultar la página man del mismo, mediante "man *comando*".

¿Cómo instalo, desinstalo o actualizo paquetes rpm?

Los paquetes rpm son archivos que llevan incluidos dentro de ellos todos los archivos que componen un determinado programa. Internamente están comprimidos, pero nosotros sólo debemos pensar en ellos en términos de Instalación, Actualización, Borrado y Consultas. Dentro del rpm van los archivos del programa a instalar, su descripción, a que directorios van a ir instalados, scripts de auto-configuración en algunos casos, etc.

La sintaxis de rpm es `rpm -acción nombre_del_paquete`

Acciones:

```
rpm -i archivo    (instalar)
rpm -e paquete    (desinstalar)
rpm -u paquete    (actualizar)
rpm -qi paquete   (pedir info)
```

Ejemplos:

```
rpm -i Par-1.50-1.i386.rpm
rpm -e Par
rpm -u Par
rpm -qi Par
```

Supongamos el archivo `programa-1.0.rpm` que no tenemos instalado y que acabamos de bajar de Internet. Procedemos a su instalación:

```
rpm -i programa-1.0.rpm
```

Tras eso el programa estará instalado en nuestro Linux y podremos ejecutarlo y usarlo normalmente. Tal vez nuestro problema es que no sabemos como se llama el ejecutable y los demás archivos de configuración que le acompañan. Para solucionar eso hacemos una consulta (query) del paquete ya instalado:

```
rpm -ql programa
```

La acción `-ql` significa "query list", y nos mostrará en pantalla la lista de archivos instalados de este programa y sus directorios destinos. Si por ejemplo deseamos ver sólo los archivos instalados en los directorios bin (los ejecutables) podemos hacer uso de `grep`, la herramienta de Linux que sólo nos mostrará aquellas líneas que contengan una

```
rpm -ql programa
```

La acción `-ql` significa "query list", y nos mostrará en pantalla la lista de archivos instalados de este programa y sus directorios destinos. Si por ejemplo deseamos ver sólo los archivos instalados en los directorios `bin` (los ejecutables) podemos hacer uso de `grep`, la herramienta de Linux que sólo nos mostrará aquellas líneas que contengan una cadena determinada:

```
rpm -ql programa | grep bin
```

Esto nos mostrará sólo los archivos de "programa" que hayan sido instalados en directorios `bin`.

Si queremos saber que hace un paquete instalado, podemos verlo con la opción "query info" (`-qi`):

```
rpm -qi programa
```

Como ejemplo, veamos la salida para el paquete `fetchmail` de Linux:

```
Name       : fetchmail                Relocations: (not relocateable)
Version    : 5.0.0                  Vendor: Red Hat Software
Release    : 1
Install date: dom 30 may 1999 16:00:12 CEST
Group      : Applications/Internet
Size       : 565413
Packager   : Red Hat Software http://developer.redhat.com/bugzilla
Summary    : A remote mail retrieval and forwarding utility.
Description:
Fetchmail is a remote mail retrieval and forwarding utility intended
for use over on-demand TCP/IP links, like SLIP or PPP connections.
Fetchmail supports every remote-mail protocol currently in use on the
Internet (POP2, POP3, RPOP, APOP, KPOP, all IMAPs, ESMTP ETRN) for
retrieval. Then Fetchmail forwards the mail through SMTP, so you can
read it through your normal mail client.
```

Si el programa no nos gusta, la desinstalación es muy sencilla:

```
rpm -e programa
```

Obviamente, no tenemos porqué instalar los programas para ver su contenido o información. Los podremos ver antes de la instalación insertando un comando `p` antes de la acción:

```
rpm -qpi archivo.rpm
rpm -qpl archivo.rpm
```

Si queremos ver la lista de RPMs instalados disponemos del comando "query all" (`-qa`):

```
rpm -qa
```

disponemos instalado. En esto caso hay 2 opciones: bien eliminar el programa anterior (-e) e instalar este nuevo (-i), o, simplemente, actualizar el programa a la versión 2.0 con el comando -U (de Update):

```
rpm -U programa-2.0.rpm
```

Por último, si al tratar de instalar un paquete rpm aparece el siguiente error:

```
Data type X not supported
```

Esto es debido a que nuestra versión de rpm es muy antigua, al menos más que el rpm que estamos tratando de instalar, y que este tiene algún tipo de compresión o elemento que nuestro rpm no entiende. Bastará entonces con actualizar nuestro ejecutable del RPM.

Cabe decir que también existen front-ends al programa rpm, es decir, programas en modo gráfico (o texto) que realizan las acciones del programa RPM mediante pulsaciones nuestras del ratón. Es el front-end el que se encarga de pasarle a RPM los parámetros correctos para que se realice la acción pedida por el usuario. Entre estos programas tenemos glint, gnorpm, purp, kpackage, xrpm, etc.

¿Cómo instalo paquetes .deb de debian?

Al igual que el formato RPM, los .DEB encapsulan todos los archivos y scripts vinculados a un determinado programa pero para la distribución Debian. Además de disponer de herramientas de instalación como dselect, se incluye la posibilidad de instalación manual mediante las diferentes opciones del programa dpkg:

Instalación de paquetes deb:

```
dpkg -i nombre_del_paquete.deb
```

Para hacer consultas:

```
dpkg -s nombre_del_paquete (admite comodines)
```

Para desinstalar:

```
dpkg -r nombre_del_paquete
```

El resto de opciones de dpkg puede verse con `man dpkg`.

El resto de opciones de dpkg puede verse con `man dpkg`.

¿Cómo se convierten paquetes entre RPM, DEB y TGZ con alien?

Alien es un programa que permite convertir un paquete binario entre los formatos rpm (Redhat, Suse), deb (Debian) y tgz (Slackware):

Conversion a .DEB:

```
alien --to-deb paquete.elquesea
```

Conversion a .RPM:

```
alien --to-rpm paquete.elquesea
```

Conversion a .TGZ:

```
alien --to-tgz paquete.elquesea
```

Existen otros flags (como `-i`) que permiten directamente tras la conversión realizar la instalación del paquete generado. Alien realiza la conversión descomprimiendo el paquete original y generando el nuevo paquete, diciéndonos el lugar del árbol de directorios donde dejará el paquete recién generado, según la distribución (por ejemplo, en Redhat en `/usr/src/redhat/RPMS/`).

¿Cómo instalo un nuevo programa?

La oferta de programas en linux y los formatos de paquete

En la red podemos encontrar una enorme cantidad de software para linux el cual podemos encontrarlo como archivos en código fuente, que han de compilarse antes de su instalación, o en formato binario, listos para ser instalados. En formato binario encontraremos los paquetes que instalan las diversas distribuciones linux y -evidentemente- el software comercial: no iban a desvelar su código fuente.

La idea fundamental que justifica la necesidad de paquetes de software para linux estriba en que:

- ⑩ Se comprueban las dependencias de funcionamiento del paquete y éste no se instala si no están instalados los necesarios para que la aplicación funcione.
- ⑩ La instalación y desinstalación es automática y no deja "basura" en el sistema al desinstalar paquetes.

Existen varios sistemas o formatos de paquetes, que fueron establecidos por ciertas distribuciones de linux y que merecen discutirse separadamente: el sistema de paquetes Debian, RedHat y Slackware

El sistema de paquetes Debian

Este sistema fue el primero en aparecer, exclusivo de Debian y muy potente: su sistema de dependencias es el más completo y flexible que existe. Lo usa la propia Debian y Corel Linux. Los paquetes debian tienen la extensión *.deb* y la herramienta de gestión de dichos paquetes es *dpkg*. Una descripción mínima de las posibilidades de *dpkg* se indica a continuación

Instalación

`-i|--install`

```
dpkg -i < nombre del paquete.deb > ;
```

por ejemplo:

```
dpkg -i icewm_0.8.12-1.deb
```

Si deseamos instalar toda una serie de paquetes que se encuentran en un directorio determinado, por ejemplo, `/usr/local/Debian/archive`, es posible hacerlo de manera recurrente con la opción `-R`:

```
dpkg -i -R /usr/local/Debian/archive
```

Sin embargo, esta opción puede no ir bien a causa de posibles dependencias entre paquetes ya que *dpkg* los va instalando conforme los encuentra, y ese quizás no sea el orden requerido para las dependencias. Mejor es entonces hacer la instalación en dos etapas con las opciones `--unpack` y `--configure`:

```
dpkg --unpack /usr/local/Debian/archive
```

Desempaqueta todos los archivos *.deb* del directorio. Si solamente queremos configurar algunos de ellos, haremos entonces

```
dpkg --configure < nombre del paquete >
```

pero si queremos instalar todos los paquetes desempaquetados del directorio haremos:

```
dpkg --pending
```

pero si queremos instalar todos los paquetes desempaquetados del directorio haremos:

```
dpkg --pending
```

o de manera más corta,

```
dpkg -a
```

Actualización

Para actualizar se usa la misma opción `-i` que en la instalación. Si por ejemplo tenemos instalado el paquete `icewm_0.8.11-1.deb` y hacemos

```
dpkg -i icewm_0.8.12-1.deb
```

en lugar de instalarse separadamente actualiza el sistema a esta última versión.

Desinstalación

```
-r|--remove
```

Esta opción borra un paquete determinado, pero no borra los archivos de configuración. Esto puede interesarnos cuando pensemos reinstalar el paquete en el futuro, pues ya tendríamos de los archivos de configuración. Así si queremos eliminar el paquete `icewm_0.8.12-1.deb` que tenemos instalado dejando sus archivos de configuración haremos

```
dpkg -r icewm
```

NOTA: nótese que sólomente hay que escribir el cabecero del paquete, no su versión ni su extensión.

```
--purge
```

En este caso el paquete se borra junto con todos sus archivos de configuración asociados. El paquete queda completamente erradicado del sistema:

```
dpkg --purge icewm
```

Consultas

- Ⓞ `dpkg -l`: da un listado de todos los paquetes instalados en el sistema, con la versión y una breve descripción
- Ⓞ `dpkg -I < nombre del paquete.deb >`: proporciona información sobre el paquete
- Ⓞ `dpkg --print-avail < nombre del paquete >`: proporciona información sobre el paquete

(dar la ruta completa: /usr/bin/wget)

Existe una utilidad que funciona como interfaz de usuario para la gestión de paquetes debian llamada *dselect*, que nos sirve cuando hay que navegar entre la multitud de paquetes de un CD debian o en los directorios remotos mediante ftp. Pero aunque se trata de una herramienta poderosa en manos del usuario medio y avanzado, es muy temida por el neófito habida cuenta de su poco intuitivo manejo. Para evitar esta reacción del usuario novel, los desarrolladores de Debian han creado la utilidad *apt*, que en la nueva liberación Debian 2.2 debería encontrarse plenamente activa junto con un propio *front-end*.

El sistema de paquetes RedHat

El formato de paquetes diseñado por RedHat, lo usan además todas las distribuciones basadas en él como Caldera Openlinux, Mandrake, Esware, Hispafuentes y también SuSE, aunque esta última no se trata de una "emanación" de RedHat. La extensión de los paquetes es *.rpm* y la herramienta de gestión de paquetes se llama también *rpm*, cuyas posibilidades se muestran a continuación:

Instalación

-i

```
rpm -i nombre_del_paquete.rpm; por ejemplo:  
rpm -i wget-1.4.5-2.rpm
```

Sin embargo, el paquete solamente se instalará si se han satisfecho las posibles dependencias con otros paquetes. Es posible forzar la instalación rechazada a causa de dependencia con otros paquetes usando la opción *-nodeps*, pero no es aconsejable.

Actualización

-U

```
rpm -U nombre_del_paquete_actualizado.rpm  
rpm -U wget-1.5.3-1.rpm
```

de este modo se desinstala el paquete obsoleto *wget-1.4.5-2* y se instala la nueva versión *wget-1.5.3-1*. Si el programa *rpm* encuentra un archivo de configuración que no ha sido cambiado por el administrador hasta el momento de la actualización, sobrescribe el nuevo archivo de configuración en él, pero si encuentra el archivo de configuración modificado por el administrador, entonces salva el archivo de configuración antiguo con

cambiado por el administrador hasta el momento de la actualización, sobrescribe el nuevo archivo de configuración en él, pero si encuentra el archivo de configuración modificado por el administrador, entonces salva el archivo de configuración antiguo con la extensión *.rpmorig* y lo reemplaza por el nuevo.

Desinstalación

-e

```
rpm -e nombre_del_paquete
rpm -e wget
```

NOTA: Nótese que sólomente hay que escribir el cabecero del paquete, no su versión ni su extensión.

Consultas

Para realizar una consulta disponemos de la orden

```
rpm -q [parámetros] [opciones] [nombre del paquete]
```

- ⊙ rpm -q -a : lista todos los paquetes instalados
- ⊙ rpm -q < nombre del paquete > : informa del nombre, versión y número de desarrollo del paquete
- ⊙ rpm -q -i < nombre del paquete > : proporciona información verbosa acerca del paquete instalado.
- ⊙ rpm -q -l < nombre del paquete > : muestra la lista de archivos correspondiente al paquete
- ⊙ rpm -q -f <archivo> : muestra el paquete al que corresponde el archivo (especificar la ruta completa: /usr/bin/wget)

Antaño (hasta la versión de RedHat 5.2), había una herramienta estupenda para la gestión de paquetes rpm mediante interfaz gráfica de usuario: se llamaba *glint*, y fue muy aplaudido por los neófitos y administradores por su facilidad para manipular los paquetes. Hoy día ya no disponemos de él, pero si se utilizan Escritorios como KDE o GNOME, podemos usar sus utilidades *kpackage* o *gnorpm*, respectivamente.

El sistema de paquetes de Slackware

En cuanto al "sistema de paquetes .tgz" de Slackware, hay cierta distancia con respecto a los ya vistos y es que no hace comprobación de dependencias, labor que corresponde al usuario o administrador. El sistema de paquetes sirve fundamentalmente para instalar o desinstalar de manera rápida los paquetes. La ventaja de esta estrategia es que los usuarios de Slackware pueden instalar los paquetes inmediatamente sin esperar a que pasen el filtro de las dependencias. Por ello, Slackware es una distribución orientada a usuarios algo avanzado, con una idea diáfana de lo que necesitan. El sistema de paquetes

installpkg

Escribiendo

```
installpkg [ opciones ] archivo.tgz
```

instalamos el paquete en Slackware. Las opciones son:

- Ⓢ-warn: Proporciona un informe de lo que ocurriría si instalamos el paquete que envía a la salida estándar.
- Ⓢ-m: Empaqueta el contenido del directorio de trabajo y sus subdirectorios en el archivo.

removepkg

Escribiendo

```
removepkg [ opciones ] archivo.tgz
```

desinstalamos el paquete del sistema completamente, eliminando además los scripts, librerías, archivos y directorios anejos. Las opciones son:

- Ⓢ-warn: Proporciona un informe de lo que ocurrir=EDA si desinstalamos el paquete y lo envía a la salida estándar.
- Ⓢ-preserve: Reconstruye el árbol de directorios del paquete en `/temp/preserved_packages/packagename` (packagename es el nombre del paquete, en nuestro caso genérico, "archivo").
- Ⓢ-copy: Es equivalente a -warn -preserve; es decir no lo desinstala pero informa y guarda el árbol de directorios.
- Ⓢ-keep: Salva los archivos temporales creados por removepkg. Útil para *debugging*.

upgradepkg

Esta utilidad permite actualizar un paquete instalado con otro especificado. Si tienen el mismo nombre, basta con hacer:

```
upgradepkg packagename, pero si el nombre es distinto hay que escribir:  
upgradepkg oldpackagename % newpackagename.
```

Otros gestores y conversión de paquetes

Las opciones: Midnight Commander y FileRunner

Otros gestores y conversión de paquetes

Las opciones: Midnight Commander y FileRunner

Existen dos herramientas que pueden ayudarnos a instalar paquetes .rpm, .deb o .tgz, mediante el uso del llamado sistema virtual de archivos. Se trata del Midnight Commander (MC) y de FileRunner (FR). Estas herramientas permiten acceder al contenido de los paquetes como si fueran directorios, explorando su interior, y vislumbrando los archivos presentes. El conocido MC, se encuentra ya integrado en GNOME y puede descargarse desde <http://www.gnome.dk/mc/index.html>.

El octavo pasajero

Sí, a veces, dada la inexistencia de un sistema estándar de paquetes hay utilidades que los desarrolladores de un sistema han producido y preparado y que no encontramos en una determinada distribución. Así, puede haber un archivo .rpm que no existe como .deb o .tgz. En estos casos existe una herramienta inestimable que permite convertir unos paquetes en otros: *Alien*.

Alien, creado por [Giovanni Quadriglio](#) no debería utilizarse intentando reemplazar paquetes importantes del sistema, como las bibliotecas compartidas, el sistema de inicio (systemv o BSD), etc, que son esenciales para el funcionamiento del mismo. Por otra parte, no se elabora del mismo modo un paquete para Debian que para RedHat. Más aún, puede haber diferencias en los paquetes .rpm dependiendo que sean de Caldera, SuSE o RedHat; así que Tengan cuidado ahí afuera.

Alien puede descargarse de <http://kitenet.net/programs/alien>, en cualquiera de los formatos conocidos. Para usar *alien* se necesitan otras utilidades. La primera, ya que *alien* está escrito en *perl*, será disponer de *perl* instalado. Para la versión actual de *alien* (6.99) nos bastará *perl* 5.004. Para convertir paquetes a *rpm* o desde *rpm*, necesitaremos tener instalado el RedHat Package Manager; para el caso de *deb*, necesitamos *dpkg*, *dpkg-deb*, *debmake*, *gcc* y *make*. Todos estos archivos "extra" que nos harán falta para las múltiples posibilidades de *alien* pueden descargarse de <ftp://ykbsb2.yk.psu.edu/pub/alien/> como un paquete llamado "alien-extra", en cualquier formato de paquete. Las fundamentales acciones conversoras de *alien* podemos extraerlas del man:

```
alien [ opciones ] paquete a convertir
```

Las opciones son *-d* para transformar cualquier formato a *.deb* (por defecto), *-r* para convertir a *.rpm* y *-t* para pasar a *.tgz* (también es posible pasar a *.slp* de Stampede con *--to-slp*).

Instalando a las bravas

A menudo, especialmente en el caso de la programación libre de código abierto (free software, source code), se dispone del código original (fuente), que generalmente se

comprimirlo se usan las utilidades *compress*, *gzip* o *bzip2* (dando lugar a los archivos *.z* o *.Z*, *.gz* y *.bz2*, respectivamente). De este modo, podemos encontrar nuestros archivos fuente como *.tar* únicamente en el caso de programas no muy grandes, pues si no hay que comprimirlos para que la gente los descargue con mayor facilidad o quepan más archivos en un medio de almacenamiento como un CDrom. Por eso es más frecuente encontrarlos con el formato *.tar.Z*, *.tar.gz* (también escritos como *.tgz* y conocidos familiarmente como "tarballs") o *.tar.bz2*. Una vez que hemos descargado alguno de estos archivos o lo hemos copiado a nuestro disco duro desde un CDrom ó un floppy, habremos de descomprimirlos y/o desempaquetarlos en un directorio para proceder a su instalación.

Ubicación de los archivos fuente

Normalmente el código fuente de los programas se sitúan en */usr/local/src*. Para que cualquier usuario ubique ahí sus propios directorios el administrador debe otorgar los siguientes permisos (como al */tmp*): **chmod 1777 /usr/local/src**

De este modo todos los usuarios pueden poner archivos ahí, pero solamente los propietarios pueden borrar sus archivos. No obstante, en otros casos, únicamente es el administrador el que se encarga de la selección e instalación del software y entonces los permisos deben ser 755 propietario root y grupo root.

Descompresión y desempaquetado simultáneo con tar

La utilidad *tar* nos permite descomprimir (caso que sea necesario) y desempaquetar los archivos. Dependiendo del formato del archivo se utilizará una u otra opción. Recomiendo que hagas *man tar* para ver el variado plumaje de sus opciones. Las más usuales son:

- ⓪z para descomprimir un archivo comprimido *.Z*, *.z* o *.gz*
- ⓪y para descomprimir un archivo comprimido *.bz2*
- ⓪x para extraer el contenido del archivo (sus archivos y directorios empaquetados)
- ⓪v (verbose) para ver lo que está sucediendo en la pantalla
- ⓪f (en último lugar y seguido del nombre del archivo fuente) para dirigir la salida de *tar* a ese archivo
- ⓪t lista el contenido del archivo

De este modo podemos efectuar las siguientes acciones:

```
tar xvf archivo.tar
tar zxvf archivo.tar.Z
tar zxvf archivo.tar.gz (ó tgz)
tar yxvf archivo.tar.bz2
```

No obstante, antes de proceder a la descompresión/dempaquetado del archivo fuente es

```
tar yxvf archivo.tar.bz2
```

No obstante, antes de proceder a la descompresión/desempaquetado del archivo fuente es de recibo hacer un `tar tzf` al archivo fuente para ver si los archivos se expanden dentro de un subdirectorio o directamente en `/usr/local/src`. En este último caso (algo raro), se debe crear un subdirectorio en `/usr/local/src` copiar el archivo `tar` a este subdirectorio y proceder a desempaquetarlo.

NOTA: Este último caso no es muy común, pero se dan algunos casos en los que el creador del archivo `tar`, queriendo o sin querer, comete el fallo de no empaquetar los archivos dentro de su propio directorio. Esto a sido causa de que a más de uno de nosotros se nos haya llenado p.ej. el directorio `/usr/local/src` de un monton de "basura" y que hayamos pasado un "ratito" limpiando dicho directorio.

Instalación

Una vez que hemos descomprimido y desempaquetado archivos y directorios, debemos buscar y leer cuidadosamente los archivos `README` e `INSTALL`, donde se nos indicará paso a paso como habrá de hacerse la instalación. En muchos casos se nos advierte de que hay que modificar el `Makefile` o escoger algunos directorios determinados para la ruta de bibliotecas ó de ciertos ejecutables o hacer algún enlace simbólico o copiar algún archivo en un directorio dado. De cualquier forma, lo más común es ejecutar lo siguiente.

configure

Este programa analiza la configuración del sistema y genera un archivo *Makefile*. Dicho archivo indica los pasos que el programa `make` ha de seguir para compilar la aplicación. En algunos casos no hay archivo `configure` y hemos de editar nosotros el `Makefile` o el `config.h` para establecer las opciones que nos interesen. No está de más echar un vistazo de todas maneras. Una vez realizado este primer paso debemos ejecutar

make

Este programa invoca al compilador siguiendo las instrucciones del archivo `Makefile`. Algunas veces habrá que hacer `make all` o `make nombre_del_programa`, como indique el archivo `README` o `INSTALL`. Mientras se compila, aparecen en la pantalla mensajes indicadores de las tareas que se están realizando. Pueden aparecer muchos *warnings*, pero pueden ignorarse, ya que no darán problemas. La compilación habrá terminado con éxito si no se detiene con algún mensaje de Error. De este modo se habrá creado el archivo ejecutable. Para instalarlo convenientemente ejecutaremos el comando `su` para transformarnos en superusuario y entonces escribiremos

make install

Makefile o hacer `./configure --prefix=/directorio_de_instalación`. En otros casos, no hay que hacer `make install` y copiamos a mano los archivos a los correspondientes directorios o hacemos un enlace simbólico a los mismos. Existen otras opciones que pueden realizarse después como `make clean`, etc, que vendrán indicadas en el archivo `README` o `INSTALL`.

NOTA: Este proceso que puede parecer muy complicado al principio, no lo es tanto cuando lo hayas hecho unas cuantas veces. Lo importante es leerse detenidamente los archivos `README` o `INSTALL` que vengan con el programa.

¿Cómo se cambian los permisos de archivos y directorios?

Lo primero que hay que decir es que para conseguir toda la información sobre los comandos involucrados en el tema de permisos puedes consultar `man chmod`, `man chown` y `man chgrp`

Información de un archivo/directorio

Cuando obtienes información sobre un archivo/directorio, existen diferentes campos que te dicen que clase de permisos el archivo/directorio tiene.

```
Ej:| [user@localhost]# ls -l
    -rwxr-x---  1 pepito depart1    4348 Nov 24 16:19 test
```

En la *primera columna* se pueden ver una serie de letras `-rwxr-x---`, estas letras nos dicen quien en el sistema, y que clases de permisos tiene el archivo `test`.

Estas letras están agrupadas en tres grupos con tres posiciones cada uno, más una primera posición que nos dice de que clase de archivo se trata (los mas normales (d) directorios, o (-) archivos de datos). En nuestro ejemplo la primera posición es (-) con lo cual el archivo `test`, es un archivo de datos (binario/ejecutable en este ejemplo).

El primer grupo de tres (rwx en nuestro caso) nos dice que clase de permisos tiene el dueño del archivo (u)(user/owner).

El segundo grupo de tres (r-x en nuestro caso) nos dice que clase de permisos tiene el grupo del archivo (g)(group).

Y el último grupo de tres (--- en nuestro caso) nos dice que clase de permisos tienen todos los demás usuarios del sistema sobre este archivo(o)(others).

r :significa permiso para leer

todos los demás usuarios del sistema sobre este archivo(o)(others).

r :significa permiso para leer
w :significa permiso para escribir
x :significa permiso para ejecutar

La *segunda columna* pepito, nos dice quien es el dueño del archivo (pepito en este caso).

La *tercera columna* depart1, nos dice cual es el grupo del archivo (depart1 en este caso).

La *cuarta columna* 4348, nos dice el tamaño del archivo.

La *quinta columna* Nov 24 16:19, nos dice cual es la fecha y hora de la última modificación.

La *sexta columna* test, nos dice cual es el nombre del archivo/directorio.

Asi pues, el archivo test de nuestro ejemplo tiene los siguientes permisos:

- ⊙ *pepito* puede leer, escribir/modificar, y ejecutar el archivo test.
- ⊙ *depart1* puede leer, y ejecutar pero no escribir/modificar.
- ⊙ Los *demás usuarios* no pueden hacer nada, ni leerlo ni modificar, ni ejecutarlo.

Como cambiar los permisos/dueño/grupo de un archivo/directorio?:

Para cambiar el dueño del archivo: `chown usuario archivo`

Para cambiar el grupo del archivo: `chgrp grupo archivo`

Para cambiar los permisos se utiliza el comando: `chmod permisos archivo`

Los permisos se pueden especificar de diferentes maneras, una serie de ejemplos, es lo mejor para comprenderlo:

```
chmod ugo+rwx test (da permisos rwx a todos, user,group,others)
chmod ugo-x test (quita permiso x (ejecucion) a todos,
user,group,others)
chmod o-rwx test (quita permisos rwx a others)
chmod u=rwx,g=rx test (da permisos rwx a user, rx a group y ninguno a
others)
```

Asi podriamos continuar con todas las posibles combinaciones de letras, es cuestión de usar la imaginación ;-)

Existe otro metodo que utiliza numeros, en vez de letras para asignar permisos, la siguiente tabla nos puede ayudar un poco a comprender esta manera:

r	w	x	VALOR DECIMAL
0	0	0	0 (000 binario es 0 en decimal)
0	0	1	1
0	1	0	2

1 1 1 7 (111 binario es 7 en decimal)

1 significa activado y 0 desactivado, o sea 101 activa r y x, y desactiva w.

Sabiendo esto solo tenemos que usar el valor decimal, un ejemplo aclarara esto.

```
chmod 750 test
```

```
da permisos rwx al usuario (7=111)
da permisos r-x al grupo (5=101)
da permisos --- a los demas (0=000)
```

¿Qué es el protocolo FTP?

El Protocolo de Transferencia de Archivos (File Transfer Protocol o, para abreviar, FTP) es uno de los pilares de Internet junto con el acceso a páginas Web (http). Durante mucho tiempo ha sido el servicio TCP/IP de Internet más utilizado, debido a que cualquier distribución Linux (así como en otros Sistemas Operativos) incluye el software necesario, muy fácil de usar y configurar, y de reducido tamaño.

Tal y como se comenta en la documentación (man ftp), "el Protocolo de Transferencia de Archivos (FTP) es un protocolo de la familia TCP/IP usado para transferir archivos entre máquinas que ejecutan TCP/IP (aunque hay programas estilo FTP para otros protocolos). Este protocolo permite transferir archivos y manipular directorios en máquinas remotas. No está diseñado para permitir el acceso a otra máquina con el fin de ejecutar programas (para eso tenemos telnet), pero resulta la mejor utilidad para manipulación de archivos."

Esto quiere decir que una máquina con un servidor de ftp dispone de una estructura de directorios con archivos y que además es capaz de permitir a cientos o miles de usuarios la recogida de archivos o envío de nuevos archivos al servidor.

Cuando hablamos de FTP estamos hablando (para explicarlo de una manera sencilla) de la descarga o subida de archivos en Internet entre 2 máquinas, y es lo que permite entrar en ftp.idsoftware.com, por ejemplo, y bajarse la última versión del ejecutable de Quake II para Linux. También nos permite, por ejemplo, entrar en nuestro directorio privado en nuestro Proveedor de Servicios de Internet para poner allí nuestra página Web con el fin de que sea accesible desde el exterior. Para hacer esto disponemos de múltiples posibilidades debido a la disponibilidad de software para ello en Linux, desde ftp hasta ncftp, wget, etc..

Así pues, no es necesario utilizar el navegador de WWW para descargar programas de

posibilidades debido a la disponibilidad de software para ello en Linux, desde ftp hasta ncftp, wget, etc..

Así pues, no es necesario utilizar el navegador de WWW para descargar programas de direcciones que comienzan por ftp://, ya que disponemos en nuestro sistema de herramientas mucho más potentes para tales menesteres: los programas de ftp.

¿Qué es el correo electrónico?

El correo electrónico es casi probablemente el servicio más útil para muchos de los usuarios de Internet, permitiendo intercambio de información casi instantáneamente. Este servicio consiste, dicho de una manera sencilla, en el envío de un bloque de información de texto (aunque puede llevar incluida información multimedia o en general cualquier otro archivo adjunto) de un remitente a un destino, de igual modo que el correo convencional. La diferencia radica en que en lugar de necesitar lapiz y papel, nuestro correo electrónico (email, de electronic mail) es generado normalmente por nuestro programa de correo como un bloque de texto a enviar con el siguiente formato:

```
From: <direccion_email_del_remitente>
To: <direccion_destino_del_email>
Subject: <tema_del_que_se_habla>
Otros: <otros_campos_de_informacion>
<texto_que_deseamos_enviar.>
```

Las direcciones de email son direcciones en servidores de correo igual que las direcciones postales lo son para la localización geográfica (por ejemplo, mi email es sromero@unix-shells.com, y un mensaje a dicha dirección llegaría a mi máquina servidora de correo de donde yo lo recogería para su lectura).

Un ejemplo de email sería el siguiente:

```
From: agonzalo@server.es
Date: Wed, 12 May 1999 22:19:24 +0200
Organization: Universidad de Sevilla
To: sromero@unix-shells.com
Subject: Una pregunta sobre configuraciones...
```

```
Hola, te quería hacer una pregunta...
¿me puedes ayudar con fetchmail?
```

Este bloque de texto se envía al servidor de correo de nuestro proveedor (o al nuestro, como ahora veremos) quien se encargará de moverlo por Internet hasta llegar al destinatario de nuestro mensaje. Podemos dividirlo en 2 partes: la cabecera del mensaje (todos los campos From, To, etc.), necesaria para el correcto envío del mismo, y el cuerpo del mensaje (el texto en sí) que es lo que deseamos enviar.

de correo son Pine, Netscape Communicator, MailX, Kmail, etc.

¿Esto no lo puedo hacer en modo gráfico?

Como regla general, se podría decir lo siguiente: *"Todo lo que se puede hacer en modo gráfico, se puede hacer también en modo texto, a base de comandos. Pero no todo lo que se puede hacer en modo texto, se puede hacer en modo gráfico"*.

Cada día aparecen más y más programas, que permiten hacer en modo gráfico, a golpe de ratón y ventanas, lo que se hace en modo texto a base de comandos. Es mas, lo único que estos programas hacen, es crear una versión gráfica que hace uso de los comandos que se utilizan en modo texto. Gestores de vaentanas como Gnome y Kde, vienen con una serie de programas gráficos que hacen esto.

Para terminar esta sección, solo decir a aquellos que quieran profundizar en el sistema, que hasta el momento, ningun programa gráfico, supera en versatilidad y potencia, a una consola de texto donde ejecutar comandos directamente.

Combinaciones especiales de teclas

- Ⓞ [Ctrl]+[Alt]+[Del]: Shutdown. Apaga el sistema de forma organizada desde un terminal texto.
- Ⓞ [Ctrl]+[Alt]+[F1]: Cambia a la primera terminal de texto.
- Ⓞ [Ctrl]+[Alt]+[Fn]: Cambia a la terminal de texto numero n (n=1, ..., 6)
- Ⓞ [Ctrl]+[Alt]+[F7]: Cambia a la primera terminal X (si se esta usando alguna)

NOTA: Si estamos en XWindow y por alguna razón nuestro entorno gráfico dejase de funcionar, siempre podriamos entrar en una terminal texto con [Ctrl]+[Alt]+[Fn], terminar los procesos que esten dando problemas y volver a las X con [Ctrl]+[Alt]+[F7].

- Ⓞ [Ctrl]+[Alt]+[Fn]: Cambia a la terminal X numero n (n=7, ..., 11)
- Ⓞ [Tab]: Autocompleta el nombre de un comando, archivo, directorio, programa, cuando trabajamos en una terminal texto. P.ej: Si vamos a usar el programa "programa_de_prueba", podemos escribir progr y pulsar [Tab], el sistema se encargará de rellenar el resto.

cuando trabajamos en una terminal texto. P.ej: Si vamos a usar el programa "programa_de_prueba", podemos escribir progr y pulsar [Tab], el sistema se encargará de rellenar el resto.

- ⑩ [ArrowUp]: (Flecha arriba) Va editando la historia de comandos que hemos escrito anteriormente en terminal texto. [Enter] para ejecutar.
- ⑩ [Shift] [PgUp]: Scroll la salida de la terminal hacia arriba, en terminal texto.
- ⑩ [Shift] [PgDown]: Scroll la salida de la terminal hacia abajo, en terminal texto.
- ⑩ [Ctrl]+c: Termina el proceso actual. Cuando lo comenzamos sin & ([localhost]\$ proceso)
- ⑩ [Ctrl]+d: Termina la terminal actual.
- ⑩ [Ctrl]+s: Para la transferencia a la terminal.
- ⑩ [Ctrl]+z: Manda el proceso actual (comenzado sin &) a "Background". Lo mismo que si comenzamos el proceso con & ([localhost]\$ proceso &).

Lista de comandos más usados

Todos los comandos citados en este apartado, pueden usarse con multitud de opciones. Para conseguir información detallada de cada comando utilizar `man comando`. A continuación tienen la manera más sencilla de utilización.

Información del sistema

- ⑩ `hostname`: Devuelve el nombre de la máquina.
- ⑩ `uptime`: Devuelve la cantidad de tiempo transcurrido desde la última vez que se arranco el sistema, la cantidad de usuarios trabajando en el sistema y el load average (carga del sistema).
- ⑩ `uname -a`: Información sobre el sistema operativo de la máquina.
- ⑩ `dmesg | more`: Imprime el "ring buffer" del kernel.
- ⑩ `free -tm`: Información sobre la cantidad de memoria disponible y usada.

- comenzando en el directorio raíz /.
- Ⓞps: Información sobre los procesos del actual usuario, que se están ejecutando.
- Ⓞps axu: Información sobre todos los procesos que se están ejecutando en la máquina.
- Ⓞtop: Información sobre el uso de cpu y memoria de los procesos del sistema.
- Ⓞcat /proc/cpuinfo: Información sobre el microprocesador.
- Ⓞcat /proc/interrupts: Información sobre las interrupciones en uso.
- Ⓞcat /proc/dma: Información sobre dma en uso.
- Ⓞcat /proc/filesystems: Información sobre los sistemas de archivos que se pueden utilizar (compilados en el kernel).
- Ⓞ/sbin/lsmmod: Información sobre los módulos en uso.
- Ⓞfinger *usuario*: Información sobre el usuario *usuario*.
- Ⓞwho: Información sobre los usuarios usando el sistema.
- Ⓞid *usuario*: Información sobre UID, GID y GROUPS del usuario *usuario*.
- Ⓞlast: Información sobre los últimos usuarios que han usado el sistema.
- Ⓞset | more: Información sobre el entorno de usuario actual.
- Ⓞping *maquina*: Para comprobar si tenemos contacto con la máquina *maquina* (*maquina*: nombre o IP)
- Ⓞ/sbin/route: Tabla de enrutamiento de nuestro sistema.
- Ⓞ/sbin/ifconfig: Información sobre los distintos dispositivos de red.
- Ⓞnetstat: Información valiosa sobre la conexión de red (Este comando tiene muchas posibilidades, ejecutar `man netstat`

Administración

- Ⓞsu: Te conviertes en administrador(root) después de introducir la clave de acceso. Ideal para realizar alguna tarea de administración sin necesidad de salir del sistema y entrar de nuevo como root.

- ⑩ `su`: Te conviertes en administrador(root) despues de introducir la clave de acceso. Ideal para realizar alguna tarea de administración sin necesidad de salir del sistema y entrar de nuevo como root.
- ⑩ `/usr/sbin/adduser usuario`: Registra y crea una cuenta de usuario.
- ⑩ `/usr/sbin/userdel usuario`: Borra la cuenta de usuario *usuario*.
- ⑩ `passwd`: Cambia la clave de acceso para el usuario actual. Root puede cambiar la clave de cualquier usuario `passwd usuario`
- ⑩ `/usr/sbin/groupadd grupo`: Crea un nuevo grupo.
- ⑩ `/usr/sbin/groupdel grupo`: Borra un grupo.
- ⑩ `more /etc/passwd`: Muestra el archivo de claves del sistema. (Si no se usa shadow password)
- ⑩ `more /etc/group`: Muestra los grupos registrados en el sistema.
- ⑩ `chmod permisos archivo/directorio`: Cambia los permisos de archivos/directorios
- ⑩ `cp archivo1 archivo2`: Copia archivo1 como archivo2
- ⑩ `ls -l` lista el contenido del directorio donde nos encontramos
- ⑩ `cd /direccion raiz/direccion hija/.....` Cambia al directorio expresado
- ⑩ `rm archivo`: Borra archivo
- ⑩ `rm -R directorio`: Borra el contenido completo (archivos/subdirectorios) de directorio
- ⑩ `mv archivo1 archivo2`: Cambia el nombre de archivo1 a archivo2
- ⑩ `mkdir directorio`: Crea un subdirectorio
- ⑩ `rmdir directorio`: Borra un subdirectorio
- ⑩ `shutdown -t3 -r now`: Reinicializa el sistema (hay que hacerlo como root).
- ⑩ `shutdown -t3 -h now`: Apaga el sistema (hay que hacerlo como root).

Glosario

- **Alan Cox:** Junto con Linus Torvalds, unos de los desarrolladores mas activos del kernel.
- **Alpha:** La computadora de arquitectura RISC (*Reduced Instruction Set Computer*) (Computadora con juego de instrucciones reducido) desarrollada por Digital Equipment Corporation.
- **Aplique:** Normalmente se trata de un pequeño programa diseñado para ser ejecutado empotrado en otro programa anfitrión.
- **Applet:** Inglés. Vea «aplique».
- **Arranque dual:** El acto de configurar un computador para que pueda arrancar más de un sistema operativo. El nombre es algo confuso, pues es posible arrancar más de dos sistemas operativos, a diferencia de lo que implica la palabra «dual». En inglés: «*Dual Boot*».
- **Arranque:** Proceso por el cual un computador comienza a ejecutar un sistema operativo cuando se le aplica la energía de alimentación. En inglés: «*bootstrap*» o más comúnmente «*boot*».

- ⑩ **Arranque:** Proceso por el cual un computador comienza a ejecutar un sistema operativo cuando se le aplica la energía de alimentación. En inglés: «*bootstrap*» o más comúnmente «*boot*».
- ⑩ **ATAPI:** Sigla de *AT Attachment Packet Interface* (interfaz de paquetes para conectar a AT). ATAPI es el protocolo mediante el cual las unidades de CD-ROM se comunican con la computadora sobre la interfaz IDE.
- ⑩ **Biblioteca:** Cuando se habla de computadores, se refiere al conjunto de rutinas que realizan las operaciones usualmente requeridas por los programas. Las bibliotecas pueden ser compartidas, lo que quiere decir que las rutinas de la biblioteca residen en un archivo distinto de los programas que las utilizan. Las rutinas de biblioteca pueden «enlazarse estáticamente» al programa, en cuyo caso se agregan físicamente las copias de las rutinas que el programa necesita. Estos binarios enlazados estáticamente no requieren de la existencia de ningún archivo de biblioteca para poder funcionar. Los programas enlazados con bibliotecas compartidas no funcionarán a menos que se instalen las bibliotecas necesarias. En inglés: «*library*».
- ⑩ **Binario:** Aunque se denomina binario al sistema de numeración en base dos que usan las computadoras, con frecuencia la palabra se refiere a la forma ejecutable de un programa. Lo contrario a «código fuente». En inglés: «*binary*».
- ⑩ **Binary:** Inglés, vea «binario».
- ⑩ **BIOS:** Sigla de *Basic Input/Output System* (sistema de entrada/salida básico).

En los sistemas compatibles con PC, el BIOS se utiliza para realizar todas las funciones necesarias para colocar en estado inicial el hardware del sistema cuando se lo conecta a la alimentación de energía. El BIOS controla el proceso de arranque, proporciona rutinas de entrada/salida de bajo nivel (de aquí su nombre) y (usualmente) permite que el usuario modifique los detalles de la configuración del hardware del sistema.
- ⑩ **Boot Diskette:** Inglés, vea «disquete de arranque».
- ⑩ **Boot:** Abreviatura de «bootstrap». Vea «arranque».
- ⑩ **Bootstrap:** Inglés, vea «arranque».
- ⑩ **Cabeza:** Cuando se refiere a unidades de disco, la cantidad de cabezas de una unidad de disco. En cada platina de una unidad de disco, hay dos cabezas -- una en cada superficie -- aunque una de las superficies no se utilice. En inglés: «*head*». Vea también «geometría».
- ⑩ **Carga del sistema:** Es una medida que nos indica la carga que están produciendo

características del anterior, cada uno consumiría el 50% de la potencia de la CPU y la carga subiría a 2.00. y así sucesivamente. La carga del sistema se puede consultar con los comandos `top` y `w`.

Ⓞ **Cilindro**: Cuando se refiere a unidades de disco, corresponde a la cantidad de distintas posiciones que pueden ocupar las cabezas de lectura/escritura sobre la platina del disco. Cuando se mira desde arriba de las platinas, cada posición de una cabeza describe un círculo imaginario con diferentes diámetros sobre la superficie de la platina, pero cuando se mira de costado, estos círculos pueden pensarse como una serie de cilindros anidados uno dentro de otro, y de allí el término. En inglés: «*cylinder*». Vea también «Geometría».

Ⓞ **CISC**: Sigla de *Complex Instruction Set Computer* (computadora con juego de instrucciones complejo). Se trata de una filosofía de diseño de computadores en la cual el procesador se diseña para ejecutar una cantidad relativamente grande de instrucciones diferentes, cada una de las cuales tarda distinto tiempo en ejecutarse (de acuerdo a la complejidad de la instrucción). Lo contrario de «RISC».

Ⓞ **CMOS**: En un principio la sigla de *Complementary Metal Oxide Semiconductor* (semiconductor de óxido metálico complementario) -- la tecnología de semiconductores que se utiliza en muchos circuitos integrados. Ahora con frecuencia designa al hardware de bajo nivel que contiene la configuración del BIOS y el reloj por hardware de la computadora.

Ⓞ **Código fuente**: El formato entendible por las personas de las instrucciones que conforman un programa. También se lo conoce como «fuentes». Sin los fuentes de un programa es muy difícil modificarlo. En inglés: «*source code*».

Ⓞ **Consola virtual**: Las consolas virtuales proporcionan múltiples «pantallas» en las cuales el usuario puede ingresar y ejecutar programas. El monitor del computador muestra una pantalla a la vez; hay una secuencia de teclas para alternar entre las distintas consolas virtuales. En inglés: «*virtual console*».

Ⓞ **Controlador de dispositivo**: Software que controla un dispositivo que está conectado a, o es parte de, una computadora. (En inglés: *device driver*).

Ⓞ **Cylinder**: Inglés, vea «cilindro».

Ⓞ **Daemon**: Inglés, ver «Demonio».

Ⓞ **Demonio**: Un demonio es un programa que funciona sin intervención humana, para cumplir una tarea determinada. Por ejemplo, `lpd` es un demonio que controla el flujo de los trabajos de impresión en una impresora.

Ⓞ **Dependencias**: Cuando se refiere a paquetes, las dependencias son requerimientos que existen entre paquetes. Por ejemplo, el paquete `foo` puede requerir archivos

flujo de los trabajos de impresión en una impresora.

⑩ **Dependencias:** Cuando se refiere a paquetes, las dependencias son requerimientos que existen entre paquetes. Por ejemplo, el paquete *foo* puede requerir archivos que son instalados por el paquete *bar*. En este ejemplo, *bar* debe estar instalado, pues sino *foo* tendrá dependencias sin resolver. Normalmente, RPM no permitirá que se instalen paquetes con dependencias sin resolver.

⑩ **Desmontaje:** El acto de revocar el acceso a un sistema de archivos. (Debe usted advertir que el programa que desmonta los sistemas de archivos se denomina `umount`.) En inglés: «*unmount*».

⑩ **Device Driver:** Inglés, vea «controlador de dispositivo».

⑩ **Dirección IP:** Las direcciones IP son el método mediante el cual se identifican los computadores individuales (o, en una interpretación más estricta, las interfaces de red de dichos computadores) dentro de un red TCP/IP. Todas las direcciones IP consisten en cuatro números separados por puntos, donde cada número está entre 0 y 255.

⑩ **Disco duro:** Un disco rígido contiene un medio magnético rotante (en forma de discos) que gira rápidamente. Hay pequeñas cabezas que flotan sobre la superficie de cada disco, y sirven para leer y escribir en el disco a medida que rota. En inglés: «*Hard Disk*».

⑩ **Disk Drive:** Inglés, vea «disco rígido».

⑩ **Disk Druid:** Disk Druid (Druida de disco) es un componente del programa de instalación de RHL que se utiliza para realizar las particiones de las unidades de disco durante el proceso de instalación.

⑩ **Diskette:** Inglés, vea «disquete».

⑩ **Disquete de arranque:** El disquete que se utiliza para arrancar distintas instalaciones RHL. En inglés: «*boot diskette*».

⑩ **Disquete de rescate:** Disquete que contiene un entorno de sistema rudimentario. Como el nombre sugiere, el disquete de rescate se utiliza normalmente en un intento de «rescatar» un sistema dañado para evitar la reinstalación total del sistema operativo. En inglés: «*Rescue Diskette*».

⑩ **Disquete para soporte PCMCIA:** El disquete necesario para las instalaciones de RHL que requieren el uso de un dispositivo PCMCIA durante la instalación. En inglés: «*PCMCIA Support Diskette*».

⑩ **Disquete suplementario:** Un disquete que se requiere en algunas clases de

mediante su uso en una unidad compatible. En inglés «*diskette*».

Ⓞ **Distribución:** Un sistema operativo (en general Linux), que se ha empaquetado para facilitar su instalación. En inglés: «*distribution*».

Ⓞ **Distribution:** Inglés, vea «distribución».

Ⓞ **Domain name:** Inglés, vea «Nombre de dominio».

Ⓞ **Driver:** Vea «Controlador de dispositivo».

Ⓞ **Dual Boot:** Inglés, vea «arranque dual».

Ⓞ **EIDE:** Sigla de *Enhanced Integrated Drive Electronics* (electrónica de unidad integrada mejorada), y es la nueva versión de la interfaz estándar integrada y otra nomenclatura para una implementación particular de interfaces IDE. Con EIDE se pueden tener discos más rápidos y de mayor capacidad; la mayoría de los sistemas que se venden hoy día utilizan EIDE.

Ⓞ **Errata:** Errata es el original en latín de «¡auch!». Cuando se detectan errores en el software, se realiza la reparación de los errores y los cambios con frecuencia se entregan como errata. RHL no es una excepción a esta regla; disponemos de una página web para las Errata en <http://www.redhat.com/errata>.

Ⓞ **Extended Partition:** Inglés, vea «partición extendida».

Ⓞ **FAQ:** Sigla de *Frequently Asked Questions* (Preguntas Frecuentes). La información acerca de Linux se presenta generalmente en forma de listas de preguntas y sus respuestas, denominadas FAQs.

Ⓞ **Fdisk:** *fdisk* es un programa de utilidad que se usa para crear, borrar o modificar las particiones en una unidad de disco. Hay que tener mucho cuidado al usar este programa, ya que, un uso inapropiado del mismo puede hacernos perder nuestra información en el disco duro.

Ⓞ **Filesystem:** Inglés, vea «Sistema de archivos».

Ⓞ **Floppy:** Término con connotaciones históricas para referirse a un disquete. Vea «disquete».

Ⓞ **Formatear:** Dar formato. El acto de escribir un sistema de archivos en una unidad de disco.

Ⓞ **Formatting:** Inglés, vea «formatear».

Ⓞ **FQDN:** Sigla de *Fully Qualified Domain Name* (Dominio completamente

- ⑩ **Formatting:** Inglés, vea «formatear».
- ⑩ **FQDN:** Sigla de *Fully Qualified Domain Name* (Dominio completamente expresado). Un FQDN es un nombre entendible por personas que incluye el nombre de la computadora y el nombre de dominio asociado a la misma. Por ejemplo, dada la computadora llamada «foo» y el nombre de dominio «bar.com», el FQDN será «foo.bar.com».
- ⑩ **FTP:** Sigla de *File Transfer Protocol* (Protocolo de transferencia de archivos). También es el nombre del programa que, tal como su nombre indica, permite copiar archivos desde un sistema a otro a través de la red.
- ⑩ **Gateway:** Inglés, vea «pasarela».
- ⑩ **Geometría:** Cuando se refiere a unidades de disco, las características físicas de su organización interna. Debe usted advertir que la unidad de disco puede informar una «geometría lógica» que es diferente de su «geometría física», normalmente para evitar las limitaciones impuestas por el BIOS. En inglés «*Geometry*». Vea además «cilindro», «cabeza» y «sector».
- ⑩ **Geometry:** Inglés, vea «geometría».
- ⑩ **GID:** Abreviatura de *Group ID* (Identificador de grupo). Por medio del GID se identifica la pertenencia de un usuario a un grupo. Los GIDs son números, aunque se almacenan nombres entendibles para personas en el archivo */etc/group*.
- ⑩ **Group:** Inglés, vea «grupo».
- ⑩ **Grupo:** El grupo es la manera de asignar derechos de acceso específicos a ciertas clases de usuarios. Por ejemplo, todos los usuarios que trabajan en el Proyecto X pueden agregarse al grupo *proyx*. Los recursos del sistema (como por ejemplo espacio en disco) que se dedican al Proyecto X se pueden configurar entonces para permitir su acceso total sólo a los miembros de *proyx*. En inglés: «*group*».
- ⑩ **Hard Disk:** Inglés, vea «disco rígido».
- ⑩ **Head:** Inglés, vea «cabeza».
- ⑩ **Hostname:** Inglés, vea «nombre de máquina».
- ⑩ **I18n:** Una «i» seguida de 18 letras, seguida de una «n», que corresponde a una abreviatura de la palabra inglesa «*internationalization*» (internacionalización). Vea «internacionalización».
- ⑩ **IDE:** Sigla de *Integrated Drive Electronics* (electrónica de unidad integrada), que denota la interfaz estándar usada para conectar fundamentalmente unidades de

usuales en las computadoras personales compatibles con PC. Estos procesadores incluyen el 80386, 80486, Pentium, Pentium Pro, Pentium II y Pentium III.

Ⓢ **Intercambio:** También se lo conoce como «espacio de intercambio» («*swap space*»). Cuando un programa necesita más memoria de la que hay disponible físicamente en el computador, la información que no se está utilizando en ese momento se puede escribir en un búfer temporal en el disco, denominado «swap», y de esa manera se libera memoria. Algunos sistemas operativos admiten el intercambio contra un archivo específico, pero Linux normalmente realiza los intercambios contra una partición dedicada al intercambio. El término «swap» está mal elegido, pues en Linux se lo usa para denotar el intercambio de páginas según la demanda («*demand paging*»). En inglés: «Swap».

Ⓢ **Internacionalización:** La práctica de diseñar y escribir programas que pueden configurarse fácilmente para interactuar con el usuario en más de un idioma. Con frecuencia se lo denomina «i18n», a causa de la cantidad de letras entre la «i» de arranque y la «n» final. En inglés: «*Internationalization*».

Ⓢ **Internationalization:** Inglés, vea «internacionalización».

Ⓢ **ISP:** Inglés, siglas de *Internet Service Provider*. Vea «psi».

Ⓢ **Kernel:** Inglés, vea «núcleo».

Ⓢ **Library:** Inglés, vea «biblioteca».

Ⓢ **LILO:** Cargador de arranque muy utilizado en sistemas Linux que se basan en procesadores compatibles con los de Intel.

Ⓢ **Linus Torvalds:** Creó Linux en 1991 mientras era estudiante universitario.

Ⓢ **Linux:** Sistema operativo completo, robusto, disponible libremente, que fue desarrollado originalmente por Linus Torvalds.

Ⓢ **Linuxconf:** Versátil programa de configuración del sistema escrito por Jacques Gelinas. Linuxconf proporciona un enfoque basado en menús para la configuración del sistema a través de distintas interfaces de usuario.

Ⓢ **Llamada al sistema:** Es una rutina que cumple una función a nivel del sistema en nombre de un proceso. En inglés: «System Call».

Ⓢ **Load Average:** Inglés, vea «carga del sistema».

Ⓢ **Logical Partition:** Inglés, vea «partición lógica».

Ⓢ **Máscara de red:** Una máscara de red es un conjunto de cuatro números separados

- ⑩ **Logical Partition:** Inglés, vea «partición lógica».
- ⑩ **Máscara de red:** Una máscara de red es un conjunto de cuatro números separados por puntos. Cada número se representa normalmente como el equivalente decimal de un número binario de 8 bits, lo que significa que cada número puede tomar valores entre 0 (todos los bits en cero) y 255 (todos los bits en uno). Cada dirección IP consiste de dos partes (la dirección de red y el número de máquina). La máscara de red se usa para determinar el tamaño de cada una de estas partes. Las posiciones de los bits en uno de la máscara se consideran parte del espacio reservado para la dirección de red, mientras que los bits que están puestos a cero se consideran parte del espacio apartado para el número de máquina. En inglés: «netmask».
- ⑩ **Master Boot Record:** Inglés, vea «Registro de arranque maestro».
- ⑩ **Memoria:** Cuando se refiere a computadores, la memoria (en general) es cualquier hardware capaz de almacenar datos para recuperarlos posteriormente. En este contexto, la memoria en general se refiere específicamente a la RAM.
- ⑩ **MILO:** Cargador que se usa generalmente para sistemas Linux basados en el procesador Alpha.
- ⑩ **Module:** Inglés, vea «módulo».
- ⑩ **Módulo:** En Linux, un módulo es un conjunto de rutinas que realizan funciones a nivel de sistema, y que pueden cargarse y descargarse dinámicamente desde el núcleo cuando sea requerido. Los módulos con frecuencia contienen controladores de dispositivos, y están fuertemente ligados a la versión del núcleo; la mayoría de los módulos construidos con una versión dada de núcleo, no se cargarán de manera apropiada en un sistema que corra un núcleo con versión distinta. En inglés: «module».
- ⑩ **Montaje:** El acto por medio del cual los sistemas de archivo se hacen accesibles a los usuarios del sistema. En inglés: «mount».
- ⑩ **Mount Point:** Inglés, vea «punto de montaje».
- ⑩ **Mount:** Inglés, vea «montaje».
- ⑩ **Mouse:** Inglés: ratón. Vea «ratón serie» y «ratón PS/2».
- ⑩ **Nameserver:** Inglés, vea «Servidor de nombres».
- ⑩ **Netmask:** Inglés, vea «máscara de red».
- ⑩ **NFS:** Sigla de *Network File System*, NFS es un método para lograr que un sistema

- a la máquina.
- Ⓢ **Nombre de dominio:** El nombre de dominio se utiliza para expresar que las computadoras pertenecen a una determinada organización. Los nombres de dominio son jerárquicos por naturaleza, y cada nivel de la jerarquía se separa de los otros niveles mediante un punto. Por ejemplo, el departamento de Finanzas de la Corporación Foo puede usar un dominio «finanzas.foo.com». En inglés: «*Domain Name*».
 - Ⓢ **Nombre de máquina:** Es una cadena de caracteres entendible para las personas que se usa para identificar una máquina computadora en particular. En inglés: «*hostname*».
 - Ⓢ **Núcleo:** La parte central de un sistema operativo, sobre la cual el resto del sistema se apoya. En inglés: «*kernel*».
 - Ⓢ **Operating System:** Inglés, vea «sistema operativo».
 - Ⓢ **Packages:** Inglés, vea «paquetes».
 - Ⓢ **PAM:** Sigla de *Pluggable Authentication Modules* (Módulos enchufables de autenticación). PAM es un sistema de autenticación que controla el acceso a RHL.
 - Ⓢ **Paquetes:** Archivos que contienen software; están escritos en un cierto formato que permite la fácil instalación y borrado del software. En inglés: «*packages*».
 - Ⓢ **Partición:** El segmento del espacio de almacenamiento de una unidad de disco que puede accederse como si fuese un disco entero. En inglés: «*partition*».
 - Ⓢ **Partición extendida:** Segmento de una unidad de disco que contiene otras particiones. En inglés: *Extended Partition*. Vea también «partición».
 - Ⓢ **Partición lógica:** Partición que existe dentro de una partición extendida. Vea también: «partición» y «partición extendida». En inglés: «*Logical Partition*».
 - Ⓢ **Partition Table:** Inglés, vea «tabla de particiones».
 - Ⓢ **Partition Type:** Inglés, vea «tipo de partición».
 - Ⓢ **Partition:** Inglés, vea «partición».
 - Ⓢ **Pasarela:** En términos de redes, se refiere al dispositivo que conecta uno o más computadores de una red a otra red. El dispositivo puede ser un hardware especializado (como un *router*), o puede ser una computadora de propósito general que se configura para actuar de pasarela. En inglés: «*Gateway*».

especializado (como un *router*), o puede ser una computadora de propósito general que se configura para actuar de pasarela. En inglés: «*Gateway*».

⑩ **PC Card:** Inglés, vea «PCMCIA».

⑩ **PCMCIA:** Sigla de *Personal Computer Memory Card International Association* (Asociación Internacional Tarjetas de Memoria para Computadoras Personales). Esta organización produce una serie de estándares que definen las características físicas, eléctricas y de software para pequeños dispositivos del tamaño de tarjetas de crédito que pueden contener memoria, modems, adaptadores de red, etc. También se las conoce como «PC Cards» (tarjetas para PC), estos dispositivos se usan principalmente en computadoras portátiles (aunque también algunos sistemas de escritorio pueden utilizar tarjetas PCMCIA).

⑩ **PCMCIA Support Diskette:** Inglés, vea «Disquete para soporte PCMCIA».

⑩ **Permisos:** El conjunto de identificadores que controlan el acceso a los archivos. Los permisos constan de tres campos: usuario, grupo y mundo. El campo de usuario controla el acceso del propietario del archivo, y el campo de grupo controla el acceso de cualquiera que concuerda con la especificación de grupo del archivo. Como el nombre implica, el campo mundo controla el acceso de cualquier otro usuario. Cada campo contiene el mismo conjunto de bits que especifican las operaciones que pueden o no realizarse, tales como lectura, escritura y ejecución.

⑩ **PLIP:** Sigla de *Parallel Line Internet Protocol* (Protocolo de Internet para líneas paralelas). PLIP es un protocolo que permite comunicaciones TCP/IP sobre el puerto paralelo de la computadora, mediante el uso de un cable especialmente diseñado.

⑩ **POSIX:** Sigla un tanto forzada de *Portable Operating System Interface* (Interfaz portable de sistema operativo). Conjunto de estándares que crecieron a partir del sistema operativo UNIX. El estándar POSIX define una interfaz portable para aplicaciones basadas en el popular sistema operativo UNIX. El principal objetivo de este estándar es la portabilidad de las aplicaciones a nivel de código fuente, mediante la unificación de las diferentes versiones del UNIX.

⑩ **Proceso:** Un proceso (en términos simplísticos en cierto modo) es una instancia de un programa en ejecución sobre un sistema Linux. En inglés: «*process*».

⑩ **Process:** Inglés, vea «proceso».

⑩ **PSI:** Siglas de *Proveedor de Sservicios Internet*. Empresa u organización que ofrece acceso a Internet a usuarios finales y corporativos.

⑩ **Punto de montaje:** El directorio bajo el cual se puede acceder a un sistema de

usa en castellano (ej.: «el directorio raíz»), mientras que en otros su traducción es desaconsejada (ej.: «el usuario root», en cuyo caso podría ser sinónimo de «superusuario» o «administrador»). Esta distinción favorece a los lectores hispanos pues quita algunas de las ambigüedades del término inglés.

⑩ **RAM:** Sigla de *Random Access Memory* (Memoria de acceso directo). La RAM se usa para mantener los programas mientras se están ejecutando, y los datos mientras se los procesa. La RAM es volátil, lo que significa que la información escrita en la RAM desaparecerá cuando se apague la alimentación de energía del computador.

⑩ **Ratón PS/2:** El ratón PS/2 toma su nombre a partir de la computadora original donde se comenzó a utilizar, la IBM PS/2. El ratón PS/2 puede identificarse fácilmente por el pequeño conector redondo en el extremo del cable.

⑩ **Ratón serie:** Un ratón serie es uno diseñado para conectarse al puerto serie del computador. El ratón serie puede identificarse fácilmente por el conector de forma rectangular que posee en el extremo de su cable.

⑩ **Rearrancar:** Recomenzar el proceso de arranque. En inglés: «reboot». Vea también «arranque».

⑩ **Reboot:** Inglés, vea «rearraancar».

⑩ **Red Hat Software:** Compañía de software sita en North Carolina. Produce y pone en el mercado software para el sistema operativo Linux, lo que incluye a Red Hat Linux.

⑩ **Registro de arranque maestro:** En inglés: «Master Boot Record» o más conocido por su sigla «MBR», es una sección del espacio de almacenamiento de la unidad de disco que se pone aparte con el propósito de guardar la información necesaria para comenzar el arranque en un computador personal.

⑩ **Rescue Diskette:** Inglés, vea «disquete de rescate».

⑩ **RISC:** Sigla de *Reduced Instruction Set Computer* (Computadora con juego reducido de instrucciones). Filosofía de diseño de computadoras en la cual el procesador está optimizado para ejecutar un número relativamente pequeño de instrucciones diferentes en una cantidad de tiempo predeciblemente pequeña.

⑩ **ROM:** Sigla de *Read Only Memory* (Memoria de sólo lectura). La ROM se usa para mantener los programas y datos que deben sobrevivir cuando se apaga el computador. Como la ROM no es volátil, los datos en la misma permanecerán sin cambios hasta la próxima vez que se encienda la computadora. Como el nombre implica, los datos no pueden escribirse con facilidad en la ROM; dependiendo de la tecnología que se usó en la ROM, la escritura puede requerir de un hardware

cambios hasta la próxima vez que se encienda la computadora. Como el nombre implica, los datos no pueden escribirse con facilidad en la ROM; dependiendo de la tecnología que se usó en la ROM, la escritura puede requerir de un hardware especial, o incluso ser imposible. El BIOS del computador se almacena en ROM.

- ⑩ **root:** (raíz) El nombre de la cuenta de ingreso que da acceso completo y total a todos los recursos del sistema. También se usa para describir el directorio denominado con «/», como en la expresión «el directorio raíz».
- ⑩ **RPM:** Sigla de *Red Hat Package Manager* (Gestionador de paquetes de Red Hat). rpm es también el nombre del programa que permite la instalación, actualización y eliminación de paquetes.
- ⑩ **SCSI:** Sigla de *Small Computer System Interface* (Interfaz de sistema para pequeñas computadoras), SCSI es una interfaz estándar para conectar una amplia variedad de dispositivos a la computadora. Los dispositivos SCSI más populares son las unidades de disco, aunque también es común encontrar unidades de cinta y «scanners».
- ⑩ **Sector:** Cuando se refiere a una unidad de disco, la cantidad de áreas de tamaño fijo (normalmente 512 bytes) que se pueden acceder mediante una cabeza de lectura/escritura, en una rotación del disco, sin que la cabeza cambie su posición. Vea también «geometría».
- ⑩ **Servidor de nombres:** En términos de redes TCP/IP, un servidor de nombres es un computador que traduce un nombre entendible por personas (como «foo.bar.com») en una dirección numérica (como «10.0.2.14»). En inglés: «nameserver».
- ⑩ **setgid:** Llamada al sistema que puede usarse para asignar el GID de un proceso. Los programas grabados con el atributo «setgid» pueden adoptar el GID del grupo al cual pertenece el archivo programa.
- ⑩ **setuid:** Llamada al sistema que se usa para asignar el UID de un proceso. Los programas grabados con «setuid» pueden adoptar el UID del usuario que es dueño del archivo programa. Esto se considera un posible problema de seguridad si el archivo es «setuid root».
- ⑩ **Shadow Password:** Normalmente, la contraseña de cada usuario se almacena en forma cifrada en el archivo */etc/passwd*. Este archivo debe poderlo leer cualquier usuario para que ciertas funciones del sistema trabajen correctamente. Sin embargo, esto significa que cualquiera puede obtener copias de las contraseñas cifradas de todos los usuarios, con lo cual resulta sencillo ejecutar un programa que adivine las contraseñas de los usuarios. Las «shadow passwords», por otro lado, almacenan la contraseña cifrada en un archivo distinto altamente protegido, lo que hace mucho más difícil el «crackeo» de contraseñas.

- Ⓢ **Sistema de archivos:** Es el método mediante el cual se almacena la información en las unidades de disco. Los distintos sistemas operativos normalmente usan diferentes sistemas de archivos, lo que dificulta el compartir los contenidos de una unidad de disco entre ellos. Sin embargo, Linux admite múltiples sistemas de archivos, lo cual hace posible la lectura/escritura de particiones dedicadas a MS-Windows, por ejemplo. En inglés: «*Filesystem*»
- Ⓢ **Sistema operativo:** Conjunto de software que controla los distintos recursos del computador. En inglés: «operating system».
- Ⓢ **SLIP:** Sigla de *Serial Line Internet Protocol* (Protocolo de Internet para líneas serie). SLIP es un protocolo que permite la comunicación TCP/IP sobre líneas serie (típicamente una conexión por módem a través de la red telefónica conmutada).
- Ⓢ **SMB:** Sigla de *Server Message Block* (Bloque de mensajes de servidor), SMB es el protocolo de comunicación que usan los sistemas operativos basados en MS-Windows para permitir los recursos compartidos a través de la red.
- Ⓢ **Source code:** Inglés, vea «código fuente».
- Ⓢ **SPARC:** Arquitectura RISC desarrollada por Sun Microsystems.
- Ⓢ **Supplemental Diskette:** Inglés, vea «disquete suplementario».
- Ⓢ **Swap:** Inglés, vea «intercambio».
- Ⓢ **System Call:** Inglés, vea «Llamada al sistema».
- Ⓢ **Tabla de particiones:** La tabla de particiones es la sección del espacio de almacenamiento de la unidad de disco que se pone aparte para definir las particiones que existen en dicha unidad de disco. En inglés: «partition table».
- Ⓢ **TCP/IP:** Sigla de *Transmission Control Protocol/Internet Protocol* (Protocolo de control de transmisión/Protocolo de Internet), TCP/IP es el nombre dado al estándar de redes de uso común en la actualidad en Internet.
- Ⓢ **Tipos de partición:** Las particiones tienen un campo que se usa para determinar el tipo de sistema de archivos que se espera que vaya a contener la partición. El tipo de partición es en realidad un número, aunque muchas veces nos referimos al tipo mediante un nombre. Por ejemplo, el tipo de partición denominado «Linux Native» es el 82. Tenga en cuenta que este número es hexadecimal. En inglés: «Partition Type».
- Ⓢ **Torvalds, Linus:** Vea «Linus Torvalds».

- Ⓢ **Torvalds, Linus:** Vea «Linus Torvalds».
- Ⓢ **UID:** Abreviatura de *User ID* (ID de usuario). Es el medio por el cual se identifica a un usuario en las distintas partes de un sistema RHL. Los UIDs son numéricos, aunque hay nombres en formato entendible por personas que se almacenan en el archivo `/etc/passwd`.
- Ⓢ **Unidad de disco:** Vea «disco rígido».
- Ⓢ **UNIX:** Conjunto de sistemas operativos del estilo de Linux que crecieron a partir de la versión original escrita por unos tipos de una compañía telefónica (¡Es broma! ¡Gracias Ken Thompson y Dennis Ritchie de los «Bell Telephone Laboratories» por su inspirado diseño de sistema operativo!).
- Ⓢ **Unmount:** Inglés, vea «desmontaje».
- Ⓢ **Virtual Console:** Inglés, vea «consola virtual».
- Ⓢ **Widget:** Representación estandarizada en pantalla de un control que el usuario puede manipular. Ejemplos de «widgets» son las barras de desplazamiento, los botones y las cajas de texto.
- Ⓢ **X Window System:** (Sistema de ventanas X) También denominado «X», esta interfaz gráfica de usuario proporciona la bien conocida metáfora de «ventanas sobre un escritorio», común a la mayoría de los sistemas hoy en día. Bajo X, los programas de aplicación actúan como clientes y acceden al servidor X que gestiona toda la actividad en pantalla. Además, las aplicaciones X pueden ejecutarse en un sistema distinto al del servidor X, lo que permite la visualización remota de las aplicaciones.
- Ⓢ **XFree86:** Implementación libre del «X Window System».