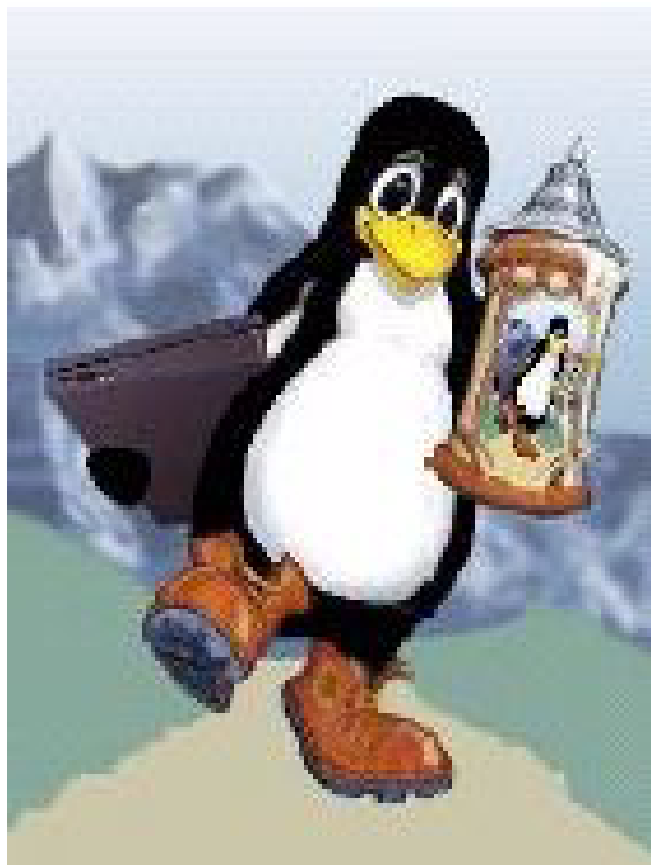


Universidad Nacional del Nordeste  
Facultad de Ciencias Exactas, Naturales y Agrimensura

Sistemas Operativos

Monografía de Adscripción

# Introducción a LINUX



Mariana Inés Kubski - L.U.: 30.524

Profesor Director: Mgter. David Luis La Red Martínez

Licenciatura en Sistemas de Información  
Corrientes - Argentina

2005



# Índice General

<b>1</b>	<b>Conceptos Básicos</b>	<b>1</b>
1.1	Introducción . . . . .	1
1.2	Antecedentes . . . . .	2
1.3	Características . . . . .	3
1.4	Ventajas Frente a Otros Sistemas Operativos . . . . .	7
1.5	Licencia . . . . .	8
1.6	Recursos Para su Desarrollo . . . . .	9
1.7	Distribuciones Comerciales . . . . .	10
1.8	Compatibilidad y Conectividad . . . . .	10
1.9	Entorno de Trabajo . . . . .	11
1.10	Usuarios y Grupos . . . . .	12
1.11	Estructura General . . . . .	13
1.11.1	Kernel . . . . .	13
1.11.2	Shell . . . . .	14
1.12	Juego de Caracteres . . . . .	14
1.12.1	Metacaracteres o Comodines . . . . .	15
1.12.2	Caracteres Especiales . . . . .	15
1.13	Sistema de Ficheros . . . . .	16
1.14	Redireccionamiento de Entrada/Salida . . . . .	17
1.14.1	Redirección Simple . . . . .	17
1.14.2	Redirección Encadenada . . . . .	18
1.14.3	Bifurcación o T . . . . .	19
1.15	Entrada y Salida del Sistema . . . . .	19
<b>2</b>	<b>Instalación</b>	<b>21</b>
2.1	Preparación . . . . .	24
2.1.1	Particiones . . . . .	24
2.1.2	Reparticionado . . . . .	25
2.2	Instalación . . . . .	27

2.2.1	Arranque . . . . .	27
2.2.2	Dispositivos y Particiones . . . . .	28
2.2.3	Creación de Particiones . . . . .	29
2.2.4	Creación del Espacio de Intercambio (Swap) . . . . .	30
2.2.5	Creación de Sistemas de Ficheros . . . . .	30
2.2.6	Instalación del Software . . . . .	31
2.2.7	Instalación de LILO . . . . .	31
2.3	Post - Instalación . . . . .	32
2.4	Posibles Problemas . . . . .	32
2.4.1	Problemas con el Arranque . . . . .	33
2.4.2	Problemas con el Hardware . . . . .	34
2.4.3	Problemas con el Software . . . . .	37
<b>3</b>	<b>Entornos Gráficos</b>	<b>39</b>
3.1	KDE . . . . .	39
3.1.1	Antecedentes . . . . .	40
3.1.2	Pantalla . . . . .	40
3.1.3	Administración de Archivos: Kfm . . . . .	41
3.1.4	Navegación por la Estructura de Directorios . . . . .	42
3.1.5	Aplicaciones . . . . .	46
3.1.6	Configuración . . . . .	50
3.1.7	Otras Aplicaciones . . . . .	53
3.2	GNOME . . . . .	55
3.2.1	Pantalla . . . . .	56
3.2.2	Aplicaciones . . . . .	57
3.2.3	Configuración . . . . .	59
3.2.4	Otras Aplicaciones . . . . .	65
<b>4</b>	<b>Sistema de Ficheros</b>	<b>69</b>
4.1	Introducción . . . . .	69
4.2	Enlaces . . . . .	71
4.3	El Camino o Path . . . . .	71
4.4	Estructura del Sistema de Ficheros . . . . .	74
4.5	Acceso al Sistema de Ficheros . . . . .	75
4.6	Permisos . . . . .	76
4.7	Comandos de Manipulación . . . . .	76
4.7.1	Listado del Contenido de Directorios: <b>ls</b> . . . . .	77
4.7.2	Creación de Subdirectorios: <b>mkdir</b> . . . . .	78
4.7.3	Borrado de Subdirectorios: <b>rmdir</b> . . . . .	78
4.7.4	Cambio de Directorio: <b>cd</b> . . . . .	78

4.7.5	Situación Actual: <b>pwd</b> . . . . .	78
4.7.6	Acceso a Unidades de Disco: <b>mount-umount</b> . . . . .	79
4.7.7	Copia de Ficheros: <b>cp</b> . . . . .	80
4.7.8	Traslado y Cambio de Nombre de Ficheros: <b>mv</b> . . . . .	80
4.7.9	Enlaces a Ficheros: <b>ln</b> . . . . .	80
4.7.10	Borrado de Ficheros: <b>rm</b> . . . . .	81
4.7.11	Características de un Fichero: <b>file</b> . . . . .	81
4.7.12	Cambio de Modo de Ficheros: <b>chmod-chown-chgrp</b> . . . . .	82
4.7.13	Espacio Ocupado en el Disco: <b>du-df</b> . . . . .	83
4.7.14	Visualización de Ficheros sin Formato: <b>cat</b> . . . . .	83
4.7.15	Visualización de Ficheros con Formato: <b>pr</b> . . . . .	83
4.7.16	Visualización de Ficheros Pantalla a Pantalla: <b>more-less</b> . . . . .	84
4.7.17	Búsqueda en Ficheros: <b>grep</b> . . . . .	85
4.7.18	Difusión de Programas y Ficheros: <b>tar-gzip</b> . . . . .	85
4.7.19	Comando de Impresión: <b>lpr</b> . . . . .	86
<b>5</b>	<b>Componentes y Distribuciones</b> . . . . .	<b>87</b>
5.1	Componentes . . . . .	87
5.1.1	Kernel GPL . . . . .	88
5.1.2	Manejadores GPL . . . . .	90
5.1.3	Librerías de Sistema y Aplicaciones . . . . .	90
5.1.4	Herramientas del Usuario . . . . .	90
5.1.5	Herramientas de Desarrollo . . . . .	91
5.1.6	X Server . . . . .	91
5.1.7	Widgets y Desktops . . . . .	91
5.1.8	Servidor WEB . . . . .	92
5.1.9	Servidor de Correo . . . . .	93
5.1.10	Servidor DNS . . . . .	94
5.2	Distribuciones Comerciales . . . . .	95
	<b>Bibliografía</b> . . . . .	<b>99</b>
	<b>Índice de Materias</b> . . . . .	<b>101</b>



# Índice de Figuras

1.1	Estructura General de los Componentes de Unix. . . . .	13
1.2	Esquema de Ejecución de Comandos. . . . .	17
3.1	KDE: Escritorios Virtuales. . . . .	41
3.2	KDE: Visualización de Archivos. . . . .	42
3.3	KDE: Barra de Navegación. . . . .	42
3.4	KDE: Aplicación Konsole. . . . .	47
3.5	KDE: Aplicación Kfind. . . . .	49
3.6	KDE: Barra de Herramientas de Kfind. . . . .	49
3.7	KDE: Menú Editor. . . . .	51
3.8	KDE: Cuadro de Diálogo. . . . .	52
3.9	KDE: Control Center. . . . .	54
3.10	GNOME: Pantalla de Inicio. . . . .	55
3.11	GNOME: Pantalla Principal. . . . .	56
3.12	GNOME: Panel. . . . .	56
3.13	GNOME: Aplicación Search Tool. . . . .	59
3.14	GNOME Control Center. . . . .	60
3.15	GNOME: Enlightenment Configuration Editor. . . . .	62
3.16	GNOME: Menú Contextual. . . . .	64
3.17	GNOME: Menú Editor. . . . .	65
3.18	GNOME: Aplicación Change Password. . . . .	67
3.19	GNOME: Aplicación System Monitor. . . . .	68
4.1	Árbol de Directorios. . . . .	72
4.2	Ejemplo de Estructura de Ficheros. . . . .	73





# Índice de Tablas

2.1	Sistema de Archivos según Sistema Operativo. . . . .	22
2.2	Manejadores de Dispositivos en Linux. . . . .	28
5.1	Componentes de Linux. . . . .	89



# Capítulo 1

## Conceptos Básicos



### 1.1 Introducción

Linux es el acontecimiento más importante en lo que a software gratuito se refiere.

Se ha convertido en el sistema operativo para los negocios, educación, y provecho personal. En los negocios ya se instala Linux en redes enteras, usando el sistema operativo para manejar registros financieros y de hospitales, un entorno de usuario distribuido, telecomunicaciones, etc. Universidades de todo el mundo usan Linux para dar cursos de programación y diseño de sistemas operativos. Y, por supuesto, entusiastas de los ordenadores de todo el mundo usan Linux en casa, para programar, entretenerse, y conocerlo a fondo.

Linux es un clónico del sistema operativo UNIX. Es una implementación de UNIX versátil, distribuida gratuitamente en los términos de la Licencia GNU.

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

## 1.2 Antecedentes

El Sistema Operativo Unix nace a finales de los años 60 en los laboratorios Bell AT & T. Ken Thompson, insatisfecho con el S.O. que utilizaba, decide escribir su propio Sistema Operativo. Inicialmente, lo escribe en lenguaje ensamblador, pero más adelante lo rescribe en un lenguaje de programación denominado B.

Paralelamente, al entrar en contacto con Unix, Dennis Ritchie junto a Ken Thompson, traducen Unix al lenguaje C.

Dada la imposibilidad de comercializarlo, AT & T decide distribuirlo con fines altruistas a Universidades, a cambio de un pago simbólico. Esta decisión tuvo dos consecuencias:

- Rápida extensión y uso en el mundo científico.
- Diversidad de versiones ya que no había quién dirigiera su desarrollo y evolución.

De estas versiones, surge Linux.

Linux es un sistema operativo gratuito y de libre distribución inspirado en el sistema Unix, inicialmente desarrollado por Linus Torvalds en la Universidad de Helsinki, en Finlandia. Se inspiraba en Minix, un pequeño UNIX desarrollado por Andy Tanenbaum. Estaba dirigido a aquellos usuarios de Minix que querían algo más.

Fue desarrollado con la ayuda de muchos programadores y expertos de UNIX a lo largo y ancho del mundo, a través de Internet.

El 5 de Octubre de 1991, Linus anunció la primera versión “oficial” de Linux, la 0.02. Ya podía ejecutar bash (el shell de GNU) y gcc (el compilador de C de GNU), pero no hacía mucho más. No había nada sobre soporte a usuarios, distribuciones, documentación ni nada parecido.

En Marzo de 1992, tras numerosas revisiones, se alcanzó la versión 0.95, reflejando la esperanza de tener lista muy pronto una versión “oficial”. (Generalmente, la versión 1.0 de los programas se corresponden con la primera teóricamente completa y sin errores).

Hoy Linux es ya un clónico de UNIX completo, con una gran variedad de características que a continuación se detallan.

## 1.3 Características

Se sabe que un sistema operativo tiene encomendada una serie de funciones diferentes.

En su papel de gestor del ordenador, el sistema operativo debe gestionar los recursos del sistema informático (procesadores, memoria, disco), entre los diferentes procesos (programas en ejecución) que compiten por ellos.

En su papel de mediador, el sistema operativo ofrece al usuario que utiliza el ordenador una especie de *máquina virtual* o *máquina extendida* más fácil de utilizar que si tuviera que acceder directamente al hardware.

Linux implementa la mayor parte de las características que se encuentran en otras implementaciones de UNIX, más algunas otras que no son habituales.

Es un sistema operativo:

- Multitarea.
- Multiusuario.
- De Planificación mixta.
- Con implementaciones de memoria virtual.

**Multitarea** dado que el sistema operativo puede atender varias tareas a la vez. A su vez estas tareas pueden provenir de un único usuario o de varios usuarios.

**Multiusuario**, donde el S.O. puede atender a un único usuario o a varios en la misma máquina, respectivamente.

**De Planificación Mixta:** la planificación define cómo se reparte el tiempo de CPU entre los diversos procesos, en caso de que en un momento dado varios procesos quieran utilizar el microprocesador. En este caso, existe una planificación concreta a base de asignar tiempos en función de prioridades. Si dos procesos tienen asignada una prioridad, se comparte el tiempo entre los dos. A los procesos poco activos se les suele dar una prioridad máxima y aquellos que exigen mucho tiempo de computación, una baja prioridad. Esto tiene sentido porque un proceso poco activo como un editor de textos consume muy pocos recursos de CPU, es decir, dentro de un intervalo de tiempo dado, el tiempo de CPU que va a usar es muy poco ya que desde que el usuario pulsa una tecla hasta que pulsa la siguiente, el ordenador ha tenido tiempo de realizar otras muchísimas tareas. Sin embargo, sería incómodo para el usuario que desde que pulsa una tecla hasta que aparece el resultado en la pantalla, pasara mucho tiempo; esa es la razón para darle máxima prioridad. En el otro extremo están los procesos que requieren mucho tiempo de computación, es decir, mucho tiempo de acceso a la CPU, como una inversión de matrices, una integral numérica, etc. Con el objeto de no saturar la máquina con estos cálculos, y debido a que habitualmente no existe una restricción temporal para acabar los cálculos, se le asigna una prioridad baja.

**Implementaciones de Memoria Virtual:** es posible hacer uso de espacio de almacenamiento en disco como si fuera memoria adicional de la que dispone el ordenador, es decir, la memoria efectiva puede ser mayor que la real. Esto se puede ver más claramente en un sistema multitarea donde las tareas de gestión son más complicadas. Si se piensa en un sistema operativo multiusuario y se supone que dos usuarios quieren imprimir simultáneamente en la misma impresora, el sistema operativo deberá decidir que uno de los usuarios puede imprimir en ese momento y que el trabajo del otro usuario quede almacenado en una cola de impresión para cuando la impresora quede libre.

Otras características son:

- Protección de memoria entre procesos, de manera que uno de ellos no

pueda colgar el sistema.

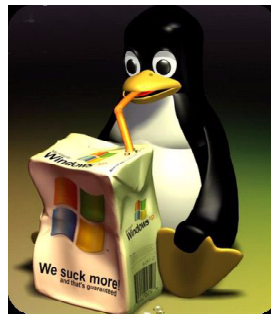
- Carga de ejecutables por demanda: Linux sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente.
- Política de copia en escritura para la compartición de páginas entre ejecutables: esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.
- Memoria virtual usando paginación a disco: una partición o un archivo en el sistema de archivos, o ambos, con la posibilidad de añadir más áreas de intercambio sobre la marcha.
- 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 éste puede a su vez ser reducido cuando se ejecuten grandes programas.
- Librerías compartidas de carga dinámica y librerías estáticas. Linux dispone de librerías compartidas enlazadas dinámicamente, lo que permite a los ejecutables ser mucho menores al enlazar el código de las librerías en tiempo de ejecución. Estas librerías DLL (Dynamically Linked Library) también permiten al programador de aplicaciones sustituir funciones ya definidas con su propio código.
- Se realizan volcados de estado 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.
- Es casi totalmente compatible con POSIX, System V y BSD a nivel fuente.
- 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.
- Control de tareas POSIX.
- Pseudo-terminales (pty's).

- Emulación de 387 en el núcleo, de tal forma que los programas no tengan que hacer su propia emulación matemática. Cualquier máquina que ejecute Linux parecerá dotada de coprocesador matemático. Por supuesto, si el ordenador ya tiene una FPU (unidad de coma flotante), será usada en lugar de la emulación.
- 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.
- Soporte para varios sistemas de archivo comunes, incluyendo minix-1, Xenix y todos los sistemas de archivo típicos de System V.
- Acceso transparente a particiones MS-DOS mediante un sistema de archivos especial.
- Un sistema de archivos especial llamado UMSDOS que permite que Linux sea instalado en un sistema de archivos DOS.
- Soporte en sólo lectura de HPFS-2 del OS/2 2.1
- Sistema de archivos de CD-ROM que lee todos los formatos estándar de CDROM.
- Appletalk disponible en el actual núcleo de desarrollo.
- Software cliente y servidor Netware disponible en los núcleos de desarrollo.
- Implementa todo lo necesario para trabajar en red con TCP/IP. Desde manejadores para las tarjetas de red más populares hasta PPP, que permite acceder a una red TCP/IP por el puerto serie (comúnmente, utilizando un Modem y la línea telefónica). Y también se han portado los clientes de TCP/IP, como FTP, telnet, NNTP (News Groups), SMTP y POP3 (E-mail) y HTTP (Web).
- Cualquier persona puede acceder a Linux y desarrollar nuevos módulos o cambiarlos a su antojo.
- Maneja discos duros de hasta 16 TeraBytes.
- Se encuentra disponible en Internet en cientos de servidores ftp y de varios distribuidores en discos flexibles o CD-ROM, alguno de estos son Caldera, Debian, Slakware, Red Hat, entre otros.



- El núcleo no utiliza ni una sola línea del código original del UNIX de AT&T o de cualquier otra fuente de propiedad comercial. Además, está legalmente protegido por la Licencia Publica GNU (GPL de sus siglas en inglés).
- El núcleo es usualmente empaquetado con varios ejecutables que comprenden un sistema operativo Unix funcional. Estos paquetes de programas son llamados “distribuciones” y vienen en muchos tamaños y arreglos.

## 1.4 Ventajas Frente a Otros Sistemas Operativos



Con el tiempo, se aprende a pensar de una forma radicalmente diferente a la que se tenía cuando se sentaba delante de otros S.O. propietarios, como el OS/2, System 7, de Mac o los Win95 o NT de Microsoft.

Con estos sistemas, uno se acostumbra a *ver* solo las opciones que el sistema le presenta, mientras que en cualquier Unix, lo que no se presenta directamente, enseguida se piensa en como hacerlo. Este cambio de mentalidad delante del sistema se convierte en un pequeño reto diario que los usuarios de Linux suelen disfrutar cada día.

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

Ahora, ¿Qué hace a UNIX tan importante?

No solo es el sistema operativo más popular para máquinas multiusuario, también es la base de la mayoría del mundo del software de libre distribución.

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

Por otro lado, OS/2 y Windows NT son sistemas propietarios. El interface y diseño están controlados por una sola corporación, y solo esa corporación puede implementar ese diseño. De alguna forma, este tipo de organización es beneficiosa: establece un estándar estricto para la programación y el interfaz de usuario distinto al encontrado incluso en la comunidad de sistemas abiertos.

OS/2 es OS/2 vaya donde vaya, lo mismo ocurre con Windows NT. Sin embargo, el interface UNIX está constantemente desarrollándose y cambiando. Varias organizaciones están intentando estandarizar el modelo de programación, pero la tarea es muy difícil.

Unix no está pensado para ser fácil de emplear, sino para ser sumamente flexible. Por lo tanto Linux no es en general tan sencillo de emplear como otros sistemas operativos, aunque se están realizando grandes esfuerzos para facilitar su uso. El futuro de Linux es brillante y cada vez más personas y más empresas (entre otras IBM, Intel, Corel) están apoyando este proyecto, con lo que el sistema será cada vez más sencillo de emplear y los programas serán cada vez mejores.

## 1.5 Licencia

Linux es distribuido libremente bajo los términos de la licencia GPL. Esta licencia permite a cualquiera trabajar en Linux. Esto significa que mientras Linux puede ser vendido en una variedad de formas, puede también ser bajado o ser copiado sin costo o restricción.

La GPL no es el único tipo de licencia de software free, y Linux se beneficia de software licenciado bajo varios otros tipos de licencias de software libremente-distribuido.

Los cuatro tipos principales son: Software del Dominio Público (Public Do-

main Software), Shareware, Derechos de Propiedad de tipo-BSD, y la Licencia Pública General (GPL).



La GPL permite toda la libertad de los otros, con un beneficio agregado, pero un beneficio para el propio proyecto: requiere que si se realiza un cambio o agregado al código GPL que se pide prestado, éste permanezca bajo los mismos términos de GPL, asegurándose que ningún desarrollador gane alguna ventaja encima de otros contribuyentes del desarrollo.

Bajo la GPL el derecho de propiedad de Linux puede ser sostenido por Linus Torvalds y otros pero ellos no tienen ningún mayor derecho para restringir el uso de él. Esta carga pequeña de falta de control restrictivo asegura que Linux y sus tecnologías relacionadas continuarán evolucionando para el beneficio igual de todos sus usuarios.

## 1.6 Recursos Para su Desarrollo

Compañías comerciales de Linux como Red Hat Software y otras, invierten directamente en Linux a través de código que sus programadores escriben y contribuyen bajo la GPL, y dinero que es donado a los equipos de desarrollo Linux más importantes como la Fundación de Software Libre y los proyectos GNU, la LDP y a los responsables por la documentación de Linux, y al equipo XFree86 y su Servidor de X.

Contribuyendo con valiosos rasgos de facilidad-de-uso al SO Linux bajo GPL, Red Hat ha hecho al Linux SO útil para un número mayor de usuarios de computadoras los cuales extienden a la comunidad de Linux. A su vez Red Hat puede hacer ventas de CD Linux, libros y aplicaciones las que ensanchan rápidamente el mercado de Linux. Pero las contribuciones de compañías como

Red Hat Software sólo representan una parte pequeña de los fondos que entran en el desarrollo de Linux.

El volumen de desarrollo de Linux es consolidado por los desarrolladores que trabajan en Linux para aplicarlo a los proyectos o aplicaciones que ellos necesitan.

Desde proyectos de super-computación de NASA a desarrollo de software de empresas como Empress Software, el trabajo es consolidado por los usuarios para su propio beneficio. El hecho de que este trabajo también beneficia de gran forma a la comunidad.

## 1.7 Distribuciones Comerciales

Una **Distribución** es un agrupamiento del núcleo del sistema operativo Linux y otra serie de aplicaciones de uso general o no tan general. En principio las empresas que desarrollan las distribuciones de Linux están en su derecho al cobrar una cierta cantidad por el software que ofrecen, aunque en la mayor parte de las ocasiones se pueden conseguir estas distribuciones desde Internet, de revistas o de amigos, siendo todas estas formas gratuitas y legales.

Las distribuciones más conocidas son *RedHat*, *Debian*, *Slackware*, *SuSe* y *Corel Linux*, todas ellas incluyen el software más reciente y empleado lo cual incluye compiladores de C/C++, editores de texto, juegos, programas para el acceso a Internet, así como el entorno gráfico de Linux: X Window.

## 1.8 Compatibilidad y Conectividad

La *compatibilidad* ha sido y será uno de los grandes éxitos de Linux, y en esto tienen mucho que ver las personas que están colaborando en el desarrollo de Linux, que trabaja habitualmente sobre distintas plataforma software, motivo este, que causa ese interés razonado por mantener la compatibilidad con cualquier otro entorno informático. Se puede decir que alrededor del 50% del software producido a nivel mundial se podría ejecutar sobre una plataforma Linux.

Entre otras cosas, se ha logrado interconectar los servidores de red Novell con cualquier servidor Linux, de forma tal que es posible montar nuevos

volúmenes desde Linux sin problemas.

Respecto a la *conectividad*, Linux puede trabajar con otros SOs sin problemas gracias a los productos desarrollados para tal fin, y uno de los mas usados es **Samba**.



Samba es un paquete que permite que un sistema Unix se integre en una red a través del protocolo NETBIOS. De esta forma sistemas Unix pueden integrarse en redes Lan Manager y compartir recursos con otros sistemas así como ofrecer recursos propios a los demás. Samba es un conjunto de aplicaciones que permiten establecer una total compartición de recursos en sistemas heterogéneos.

## 1.9 Entorno de Trabajo

Al contrario que otros sistemas operativos, por defecto el trabajo con Linux no se realiza de una forma gráfica, sino introduciendo comandos de forma manual. Linux dispone de varios programas que se encargan de interpretar los comandos que introduce el usuario y realiza las acciones oportunas en respuesta. Estos programas denominados *Shell* son el modo típico de comunicación en todos los sistemas Unix incluido Linux. Para muchas personas el hecho de tener que introducir los comandos de forma manual les puede parecer intimidante y dificultoso, aunque como se verá más adelante los comandos de Linux son relativamente simples y muy poderosos.

No obstante, casi todas las distribuciones más recientes incluyen el sistema *X Window*, el cual es el encargado de controlar y manejar la interfaz de usuario. X Window es mucho más poderoso que otros entornos similares como Microsoft Windows, puesto que permite que el usuario tenga un control absoluto de la representación de los elementos gráficos.

## 1.10 Usuarios y Grupos

Linux es un sistema operativo multitarea y multiusuario. Por lo tanto, se deben establecer ciertos mecanismos de tal manera que, simultáneamente, se protejan los datos de un usuario frente a otros y éstos puedan ser compartidos en caso necesario.

Existe un mecanismo de permisos asociados a cada fichero. Este mecanismo permite que los ficheros y directorios pertenezcan a un usuario en particular, pero también permite que los ficheros sean compartidos entre usuarios y grupos de usuarios. El comportamiento por defecto en la mayoría de los sistemas es que todos los usuarios pueden leer los ficheros de otro usuario, pero no pueden modificarlos o borrarlos.

Cada usuario, perteneciente a un grupo de usuarios, tiene asociado un nombre, una palabra clave o password, un directorio y un proceso de arranque.

Existen diferentes categorías de usuarios en función de sus privilegios:

- Superusuario o root: es el administrador del sistema. Tiene todos los privilegios.
- Usuarios normales: el resto de usuarios que pertenecen a distintos grupos, los cuales pueden tener una serie de propiedades comunes.
- Usuarios especiales: asignados a tareas específicas por el sistema, generalmente de información o manejo de aplicaciones ya instaladas de uso común a usuarios externos o internos.

Los permisos que llevan asociados todos los ficheros y directorios se clasifican en:

- Lectura (read, r).
- Escritura (write, w).
- Ejecución (execute, x).

Estos permisos se pueden asignar al propio usuario, al grupo, y al resto.

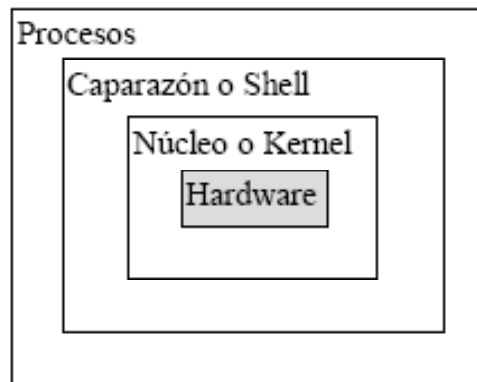


Figura 1.1: Estructura General de los Componentes de Unix.

## 1.11 Estructura General

Los diversos componentes del sistema operativo pueden verse gráficamente en la Fig. 1.1 de la pág. 13. Ellos son:

- Núcleo o Kernel: comprende un 5-10% del código total.
- Caparazón o Shell: actúa como intérprete de comandos.
- Programas de utilidad.

### 1.11.1 Kernel

Es el corazón del sistema ya que posee un gran número de funciones de sistemas que son invisibles para el usuario casual, llamadas *system calls*. Se carga al encender la máquina y funciona hasta que ésta es apagada. Realiza todas las labores propias de un sistema operativo, y ellas son:

- Planificar, coordinar y gestionar la ejecución de los procesos. Para ello, hace uso de las prioridades asignadas a cada proceso y utiliza algoritmos específicos para repartir el tiempo entre los diversos procesos que compiten por él.

- Dar servicios del sistema, como entrada/salida y gestión de ficheros.
- Manejar las operaciones dependientes de hardware, es decir, realiza las funciones demás bajo nivel de manera que se oculten al usuario.

Un kernel típico puede constar de unas 20.000 líneas de código de las cuales un 70-80% está escrito en C y el resto depende de la máquina.

### 1.11.2 Shell

Desde el punto de vista del usuario, actúa como un intérprete de comandos. Es un programa que siempre está en ejecución, procesa comandos para que el usuario interactúe con el sistema operativo. El Shell lee las órdenes suministradas, las decodifica y comunica al núcleo para realizar la acción especificada.

Se distingue por el prompt `%>`.

Permite la programación de programas de comandos (o shell scripts).

Prácticamente, todas las órdenes son programas ejecutables que el Shell busca en el sistema de ficheros, siguiendo el orden especificado en la variable global PATH. Existen varios tipos de shells, según la versión utilizada:

- Bourne shell (System V, Xenix).
- C shell (Berkeley).
- Korn shell (Ambos).

El shell puede constar, en total, de unas 200.000 líneas de código en C.

## 1.12 Juego de Caracteres

- Letras minúsculas: a, b, c, d, e, f, g, h, i, j, k, l, m, n, ñ, o, p, q, r, s, t, u, v, w, x, y, z.
- Letras mayúsculas: A, B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, O, P, Q, R, S, T, U, V, W, X, Y, Z.



- Números: 0,1, 2, 3, 4, 5, 6, 7, 8, 9.
- Caracteres especiales y metacaracteres.

Se debe destacar que la misma letra en minúscula y en mayúscula representa distinto carácter. Los comandos, en particular, suelen estar en minúscula. Asimismo, los caracteres especiales no deben formar parte del nombre ya que tienen un significado especial para el intérprete de comandos.

### 1.12.1 Metacaracteres o Comodines

- `*` Es equivalente a cualquier cadena de caracteres.
- `?` Es equivalente a cualquier carácter.

Se suelen utilizar para hacer referencia a varios archivos con un sólo comando, por ejemplo, si se quiere borrar todos los ficheros fuente en Pascal que empiezan por la letra `b`, en vez de borrar uno a uno, se debe colocar: `b*.p`. De la misma manera, si lo que se quiere es borrar aquellos cuya segunda letra sea una `b`, se debe escribir `?b*.p`.

### 1.12.2 Caracteres Especiales

Los caracteres especiales son:

- `;` Equivale a un retorno de carro en la línea de órdenes.
- `[]` Engloba grupos de caracteres, separados por coma o por guión.
- `<>` Redireccionan la entrada/salida, respectivamente.
- `&` Envía un proceso en “background”.
- `|` Para construir tuberías o “pipes”.

## 1.13 Sistema de Ficheros

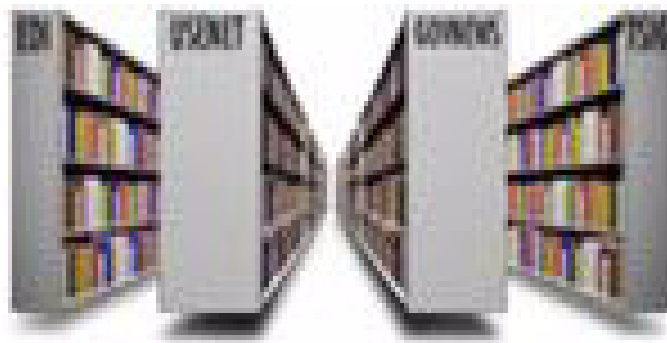
Linux emplea un sistema de ficheros jerárquico de directorios-ficheros.

No existe, a nivel de usuario, el concepto de volumen, ni de dispositivo físico. Es decir, el usuario no sabe en que disco están los ficheros que está utilizando.

Un fichero es un conjunto de información al que se le da un nombre. Existen tres tipos de ficheros:

- Ordinarios: son cadenas de bytes terminadas con `<ctrl>D` (esto significa fin de fichero). Pueden ser texto, objetos, ejecutables, bibliotecas de módulos, entre otros.
- Directorios: contienen nombres de ficheros y su dirección física. Puede pensarse en ellos como carpetas que contienen ficheros y directorios.
- Especiales: están asociados a dispositivos de entrada/salida. Contienen referencias a los drivers. Pueden ser de tipo *bloque* (apuntan a dispositivos tipo disco) y *character* (apuntan a dispositivos como terminales, impresoras, etc). Por convenio, residen en el directorio `/dev`.

Todo lo relacionado con este tema se profundiza en el Capítulo 4: Sistema de Ficheros.



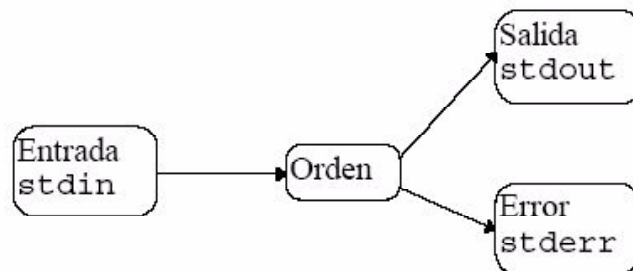


Figura 1.2: Esquema de Ejecución de Comandos.

## 1.14 Redireccionamiento de Entrada/Salida

Muchos comandos UNIX toman sus datos de entrada de la llamada entrada estándar y envían sus resultados a la salida estándar, a menudo abreviadas como `stdin` y `stdout` respectivamente. Usualmente el sistema está configurado de forma que la entrada estándar sea el teclado y la salida estándar la pantalla.

La ejecución de un comando típico responde a un esquema similar al que se visualiza en la Fig. 1.2 de la pág. 17.

En dicha figura se observa que el comando, si necesita algún dato de entrada, va a esperar que éste sea introducido a través de teclado, que es la entrada por defecto en el sistema operativo. Si el comando ha de devolver alguna información respecto al resultado conseguido por su ejecución, lo enviará al dispositivo de salida por defecto que es la pantalla. De la misma manera, posibles errores en la ejecución del comando saldrán por el dispositivo de errores por defecto, que también es la pantalla.

Este comportamiento por defecto de los comandos (entrada por teclado, salida por pantalla) puede ser modificado a través de lo que se denomina *Redireccionamiento de Entrada / Salida*.

### 1.14.1 Redirección Simple

El intérprete de comandos permite redireccionar la salida estándar a un fichero usando el símbolo `>`. Entonces, se puede variar la entrada/salida estándar

utilizando el formato:

```
comando <entrada >[>]salida 2>[>]errores
```

Los corchetes indican que lo que contienen es opcional, donde:

- **entrada** fichero sustituye a **stdin**.
- **salida** fichero sustituye a **stdout**.
- **errores** fichero sustituye a **stderr**.

Los ficheros destino pueden ser alguno de los estándar. En este caso, se especifican con los símbolos **&1** (**stdout**), **&2** (**stderr**). Los ficheros de salida y entrada deben ser diferentes.

### 1.14.2 Redirección Encadenada

Ahora, ¿qué sucede si los datos que se quieren ordenar provienen de la salida de otro comando?. Es lo que se denomina *Redireccionamiento Encadenado*, *Tuberías* o *Pipes*.

El uso de *pipes* es otra característica del intérprete de comandos que permite conectar una cadena de comandos, de manera que la **stdout** del primero es enviada directamente a la **stdin** del segundo y así sucesivamente. Para crear un pipe se utiliza el símbolo **|**. Mediante este símbolo se pueden empalmar tantos comandos como se desee.

Su formato es el siguiente:

```
nombre comando salida | nombre comando entrada
```

Resumiendo: la salida estándar de un comando de la tubería actúa como entrada estándar del siguiente.

No tiene sentido redireccionar la entrada, salvo en el primer comando. Tampoco tiene sentido redireccionar la salida, salvo en el último comando.

### 1.14.3 Bifurcación o T

A veces interesa que la salida de un comando, además de redirigirse a un determinado fichero, se bifurque también hacia la terminal, con objeto de observar inmediatamente el resultado. Esto se consigue con el operador **tee**, que se puede emplear de la siguiente forma:

```
ls | tee file
```

donde la salida de **ls** se bifurca hacia la terminal y hacia file. Si se desea que la salida de este comando se añada al final de file, se debe utilizar la opción **-a**:

```
ls | tee -a file
```

## 1.15 Entrada y Salida del Sistema

En cualquiera de los casos en la pantalla aparecerá (más o menos) lo siguiente:

**Login:** se debe teclear el nombre del usuario.

**Password:** se debe teclear la contraseña, que por razones de seguridad no se visualiza en la pantalla. Se debe cumplir con ciertas condiciones tales como:

- Contener al menos seis caracteres.
- Contener al menos un carácter numérico o especial y dos alfabéticos.
- Ser diferente del nombre de login.

La primera vez que se accede al sistema la contraseña empleada será la proporcionada por el administrador del sistema.

Para terminar la sesión de trabajo en Linux, dependiendo de si se está en modo gráfico o de texto, se debe:

En modo texto:

- Presionar las teclas **<ctrl> d**.
- Escribir el comando **exit**.

En modo gráfico:

- La salida de X Window depende del gestor de ventanas que se esté ejecutando.

En los próximos capítulos se profundizan cada uno de los conceptos desarrollados aquí.

## Capítulo 2

# Instalación



Al contrario de lo que ocurre con Microsoft Windows, la instalación de Linux no es un proceso sencillo, dado que Linux permite el control y la personalización de una cantidad mayor de parámetros y opciones. Pese a todo antes de proceder a instalar Linux es necesario tener en cuenta una serie de aspectos fundamentales.

El primero de ellos es leer la información que contiene el CD de la instalación, esta información puede aparecer de dos formas distintas, los llamados HOWTO o en forma de manuales desarrollados para la distribución.

Otro de los conceptos principales a tener en cuenta antes de la instalación es el de partición. Cada sistema operativo organiza la información de los ficheros que contiene de forma diferente, utilizando cada uno su propio sistema de archivos. Esto en general impide que se puedan instalar varios sistemas operativos mezclados en un mismo disco. Para solucionar este problema exis-

ten las llamadas particiones con las que se divide un determinado disco duro de forma que pueda contener ambos sistemas de archivos. Como referencia, la tabla 2.1 de la pág. 22 indica el nombre del sistema de archivos de diferentes sistemas operativos:

Sistema Operativo	Sistema de Archivos
MS-DOS	FAT
MS Windows 95	VFAT
MS Windows 95 OSR2 y Windows 98	FAT32
MS Windows NT	NTFS
IBM OS/2 Warp	HPFS
Linux	Ext2

Tabla 2.1: Sistema de Archivos según Sistema Operativo.

Los problemas fundamentales al instalar Linux provienen de que en la mayor parte de las ocasiones el usuario desea conservar Windows y todos los programas para este sistema. En la actualidad existen varias distribuciones que permiten la instalación de Linux en un sistema de archivos de Windows, bien en lo que se denomina un disco imagen, o directamente en el sistema de archivos. No obstante para obtener un buen rendimiento es preferible instalar Linux sobre una partición diferente de la de Windows empleando el sistema de archivos propio de Linux.

En la actualidad las distribuciones de Linux incluyen una pequeña utilidad llamada *fips* que permite dividir el disco duro sin perder información. Antes de utilizar *fips* es muy importante leer la información que se adjunta con el programa. Tras emplear *fips* se habrá reducido el tamaño de la partición de Windows y se habrá creado una nueva para ser utilizada durante la instalación de Linux. Durante el proceso de instalación de Linux habrá que borrar la partición creada con *fips* y sustituirla por las particiones que va emplear Linux.

Otro de los puntos a tener en cuenta es cómo se quiere arrancar Linux si existe Windows. La forma más sencilla es instalar LILO, un programa que se encarga de arrancar ambos sistemas operativos según lo que indique el usuario al arrancar la máquina. Los problemas surgen si hay que reinstalar Windows nuevamente, puesto que éste sistema operativo asume el control de la PC y el sistema de arranque eliminando LILO (e impidiendo arrancar Linux). Otra forma muy sencilla es emplear el disquete de arranque que se crea durante la instalación. Por último, otra manera de ejecutar Linux es creando un menú



de arranque en Windows y empleando el programa LoadLin desde MS-DOS. Para más información sobre este último método de arranque se recomienda la lectura del mini-HowTo loadlin que suele acompañar a la documentación de Linux.

No hay una única distribución oficial del software de Linux; existen muchas distribuciones, cada una de las cuales sirve a un propósito particular y una serie de objetivos. Estas distribuciones están disponibles por FTP anónimo en Internet, en BBS de todo el mundo, y en CD-ROM.

Aquí se presenta una visión general del proceso de instalación, aunque ésta varía según la distribución que se utilice. Los pasos básicos son:

1. Reparticionar el disco duro, con el fin de reservar espacio para Linux.
2. Arrancar la instalación de Linux, cada distribución incluye algo para arrancar inicialmente e instalar el software, usualmente un diskette de arranque. Al arrancar de esta manera, se iniciará un programa de instalación para el resto del software, o bien se podrá seguir instalando a mano.
3. Crear las particiones para Linux. Después de reparticionar el disco para reservar espacio para Linux, se deben crear particiones de Linux en dicho espacio. Esto se realiza con el programa fdisk.
4. Crear los sistemas de ficheros y el espacio de intercambio. Se debe crear uno o más sistemas de ficheros, utilizados para guardar los ficheros en las particiones recién creadas. Además, si se piensa usar espacio de intercambio ("*swap*"), hay que crear dicho espacio en una de las particiones para Linux.
5. Instalar los programas en los sistemas de ficheros.

La mayoría de las distribuciones de Linux proporcionan un programa de instalación que conduce en cada paso de la instalación, y automatiza algunos de esos pasos.

Instalar Linux no es difícil, pero hay que recordar continuamente muchos detalles. Conviene llevar un registro de todos esos detalles para poder probar de otra forma si algo no va bien.

## 2.1 Preparación

### 2.1.1 Particiones

En general, los discos duros se encuentran divididos en particiones, donde cada partición corresponde a un sistema operativo. Si ya se tiene otro software instalado en el sistema, puede ser necesario cambiar el tamaño de las particiones con el fin de reservar espacio para Linux. En el espacio reservado se crearán una o más particiones para almacenar el software de Linux y el espacio de intercambio.

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

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

Existen tres clases de particiones:

- Primarias.
- Extendidas.
- Lógicas.

De éstas, las más usadas son las primarias. Sin embargo, debido al límite del tamaño de la tabla de particiones, sólo se pueden tener hasta cuatro particiones primarias en un disco.

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

### 2.1.2 Reparticionado

¿Qué espacio necesita Linux?

En los sistemas UNIX, los ficheros se almacenan en un sistema de ficheros. Básicamente esto es una zona del disco duro formateado para almacenar ficheros. Cada sistema de ficheros se encuentra asociado con una parte específica del árbol de directorios; por ejemplo, en la mayoría de los sistemas, existe un sistema de ficheros para todos los ficheros del directorio `/usr`, otro para `/tmp`, etc. El sistema de ficheros raíz es el principal, que corresponde con el directorio raíz, `/`.

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

Antes de instalar Linux, se deben preparar sistemas de ficheros para almacenar el software de Linux. Por lo menos se debe tener un sistema de ficheros (el sistema de ficheros raíz), y una partición reservada a Linux. La mayoría de los usuarios de Linux optan por almacenar todos sus ficheros en el sistema de ficheros raíz, dado que es mas fácil gestionarlos que teniendo diferentes sistemas de ficheros y particiones.

¿Por qué usar más de un sistema de ficheros?

Lo más habitual es por seguridad; si, por alguna razón, uno de los sistemas de ficheros resulta dañado, los otros normalmente no resultarán afectados. Por otro lado, si se almacenan todos los ficheros en el sistema de ficheros raíz, y por alguna razón resulta dañado, se pueden perder todos los ficheros de una vez. Otra razón para utilizar varios sistemas de ficheros es repartir el almacenamiento entre varios discos duros.

En resumen, Linux requiere por lo menos una partición, para el sistema de ficheros raíz. Si se desea crear varios sistemas de ficheros, se necesitará una partición por cada sistema de ficheros. Algunas distribuciones de Linux crean particiones y sistemas de ficheros de forma automática.

Otra cuestión a considerar cuando se deciden las particiones es el *espacio de intercambio* o *swap*. Si se desea usar espacio de intercambio en Linux, se tienen dos opciones:

1. Usar un fichero de intercambio que existe dentro de uno de los sistemas

de ficheros de Linux. Entonces se crea el fichero de intercambio para usarlo como RAM virtual una vez instalado el software.

2. Crear una partición de intercambio, una partición reservada exclusivamente como espacio de swap. Es la opción más utilizada.

Cada fichero o partición de intercambio puede ser de hasta 16 megabytes. Si se desea tener más de 16 megabytes de swap, se pueden crear varios ficheros o particiones de intercambio.

También es necesario conocer el espacio requerido para cada partición. El tamaño de los sistemas de ficheros del sistema Linux depende en gran parte de qué software se quiera instalar en él y de la distribución de Linux que se esté utilizando. Esta información se encuentra en la documentación de la distribución.

El tamaño de las particiones de swap depende de la RAM virtual que se necesite. Lo típico es crear una partición de intercambio del doble de espacio de la RAM física.

El programa utilizado para hacer particiones es *fdisk*. Cada sistema operativo tiene su propia versión de este programa; por ejemplo, bajo MS-DOS, se activa con el comando FDISK.

No se debe modificar o crear particiones para otros sistemas operativos (incluyendo Linux) utilizando FDISK bajo MS-DOS. Solo pueden modificarse particiones de cada sistema operativo con la versión de fdisk correspondiente a ese sistema; por ejemplo, se crearán las particiones para Linux utilizando el programa fdisk que viene con Linux.

Para borrar una partición, se debe seleccionar la opción “Eliminar partición o unidad lógica DOS”. Luego, especificar el tipo de partición que se desea borrar (primaria, extendida o lógica) y el número de la partición.

Para crear una nueva partición para MS-DOS, se debe seleccionar la opción de FDISK “Crear partición DOS o unidad lógica DOS”. Luego, especificar el tipo de partición (primaria, extendida o lógica) y el tamaño (en megabytes).

Después de hacer esto mediante FDISK, se debe salir del programa y reformatear las nuevas particiones.

## 2.2 Instalación

Después de modificar las particiones para reservar espacio a Linux, se está en condiciones de instalar el software. Los pasos a seguir son:

1. Arrancar con el dispositivo de arranque de Linux (disquete).
2. Ejecutar fdisk bajo Linux para crear las particiones de Linux.
3. Ejecutar mke2fs y mkswap para crear los sistemas de ficheros y el espacio de intercambio.
4. Instalar el software de Linux.
5. Instalar el cargador LILO en el disco duro, o crear un disco de arranque con el fin de arrancar el nuevo sistema Linux.

Uno o más pasos de estos pueden estar automatizados por los programas de instalación, según la distribución que se esté utilizando.

### 2.2.1 Arranque

El primer paso es iniciar el computador con el dispositivo de arranque de Linux, que suele ser un disco “boot” que contiene un pequeño sistema Linux. Tras arrancar con el floppy, se presenta un menú de instalación que conduce en el proceso de instalación. En otras distribuciones, aparece un prompt de login al arrancar. La documentación que viene con la correspondiente distribución explicará que se necesita para arrancar Linux.

La mayoría de las distribuciones de Linux utilizan un disquete de arranque que permite introducir parámetros del hardware en tiempo de arranque, para forzar la detección de los dispositivos.

El prompt de arranque se muestra siempre que se arranca con el disquete. Según la versión, es necesario mantener pulsadas las teclas **shift** o **ctrl** mientras se arranca. Si todo sale bien, debería verse el prompt **boot:** y tal vez otros mensajes.

### 2.2.2 Dispositivos y Particiones

Bajo Linux, los dispositivos y las particiones tienen nombres muy distintos a los utilizados en otros sistemas operativos. Los manejadores de dispositivos, que se encuentran en el directorio `/dev`, se usan para comunicar con los dispositivos del sistema. Por ejemplo, al ratón se lo puede acceder a través del manejador `/dev/mouse`. Las disqueteras, discos duros y particiones tienen cada uno un manejador propio. La tabla 2.2 de la pág. 28 lista los nombres de diversos manejadores.

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

Tabla 2.2: Manejadores de Dispositivos en Linux.

Las particiones de `/dev/hda` son `/dev/hda1`, `/dev/hda2`, etc. Sin embargo, los dispositivos SCSI se nombran con `/dev/sda`, `/dev/sdb`, etc., y las particiones con `/dev/sda1`, `/dev/sda2`, etc.

### 2.2.3 Creación de Particiones

Después de arrancar el disquete, se debe ejecutar el comando `fdisk` tecleando:

`fdisk <drive>`, donde `<drive>` es el nombre de dispositivo con el que Linux identifica el disco duro donde se quiere realizar las particiones.

Se puede teclear `m` para obtener una lista de opciones.

Command: `m`

- `a`: toggle a bootable flag.
- `d`: delete a partition.
- `l`: list known partition types.
- `n`: add a new partition.
- `p`: print the partition table.
- `q`: quit without saving changes.
- `t`: change a partition's system id.
- `u`: change display/entry units.
- `v`: verify the partition table.
- `w`: write table to disk and exit.
- `x`: extra functionality.

El comando `n` se usa para crear una nueva partición. Para salir de *fdisk* sin salvar cambios, se debe utilizar el comando `q`. Para salir escribiendo los cambios en la tabla de particiones, se debe utilizar el comando `w`. Lo primero que conviene hacer es visualizar la tabla de particiones actual y anotar los datos, para referencias posteriores. Usar el comando `p` para esto.

Muchas distribuciones requieren que se utilice el comando `t` en *fdisk* para cambiar el tipo de la partición elegida para el intercambio a “Linux swap”. Se puede usar el comando `L` para ver una lista de tipos de particiones conocidas, y luego `t` para establecer el tipo de la partición de intercambio a “Linux swap”.

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

Las nuevas versiones de `fdisk` cambian de forma automática esta información en el núcleo, con lo que no es necesario rearrancar. Pero lo más seguro es volver a arrancar tras crear las particiones.

#### 2.2.4 Creación del Espacio de Intercambio (Swap)

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

El comando utilizado para preparar una partición de intercambio es *mkswap*:

`mkswap -c <partition> <size>`, donde `<partition>` es el nombre de la partición de swap y `<size>` es el tamaño de la partición, en bloques. La opción `-c` indica a `mkswap` que compruebe si hay bloques erróneos en la partición mientras la crea.

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

#### 2.2.5 Creación de Sistemas de Ficheros

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

Hay varios tipos de sistemas de ficheros disponibles en Linux. Cada tipo de sistema de ficheros tiene su propio formato y características (como longitud del nombre de los ficheros, tamaño máximo, etc).

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



permite hasta 256 caracteres en los nombres de los ficheros y tamaños de éstos de hasta 4 Terabytes. En el Capítulo 4: Sistema de Ficheros se verán los diferentes tipos disponibles en Linux.

Algunas versiones crean los sistemas de ficheros de forma automática. Para crear un sistema tipo *ext2fs* se debe utilizar el comando:

`mke2fs -c <partition> <size>`, donde `<partition>` es el nombre de la partición, y `<size>` es el tamaño de la partición en bloques.

### 2.2.6 Instalación del Software

Finalmente, lo más interesante. Cada distribución tiene una forma distinta de hacerlo. La mayoría tiene un programa que guía paso a paso en este proceso.

Algunas distribuciones ofrecen diversos mecanismos para instalar el software. El método exacto para instalar el software de Linux difiere en gran parte según la distribución. Por lo tanto, se puede afirmar que instalar el software de Linux es autoexplicativo.

### 2.2.7 Instalación de LILO

Cada distribución proporciona mecanismos para arrancar Linux cuando ya esté instalado en el sistema. En la mayoría de los casos se creará un disquete “boot” que contiene el núcleo de Linux configurado para usar con el sistema de ficheros raíz creado. Se deberá arrancar Linux desde ese disquete y tras el arranque se pasará el control al disco duro.

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

Con el fin de instalar LILO correctamente, se necesita conocer bastante información acerca de la configuración del disco: qué particiones contiene cierto sistema operativo, cómo arrancar cada sistema operativo, etc.

La mayoría de las distribuciones, cuando se instala LILO, tratan de elegir la mejor configuración para éste. Aunque no es lo habitual, la instalación

automatizada de LILO puede fallar, dejando el registro de arranque maestro del disco inservible. En concreto, si se utiliza el Boot Manager de OS/2, no se deberá instalar LILO mediante el procedimiento automático, para ello habrá que seguir instrucciones especiales.

## 2.3 Post - Instalación

Luego de haber completado la instalación de Linux, en la mayoría de los casos, se debe poder arrancar el sistema, entrar como root, y comenzar a explorar el sistema.

Llegado este punto es conveniente saber como rearrancar y apagar el sistema cuando se esté usando. Nunca se debe rearrancar o apagar el sistema Linux presionando el interruptor de reset o pulsando a la vez el tan conocido `ctrl-alt-del`.

Como en la mayoría de sistemas UNIX, Linux lleva una cache de disco en memoria, lo que aplaza la escritura de los datos. Es por ello que si se rearranca el sistema sin apagarlo “limpiamente”, se pueden corromper datos en las unidades, causando un daño impredecible.

La forma más fácil de apagar el sistema es usar el comando shutdown:

```
# shutdown -r now
```

Después de que haber explorado y usado el sistema, hay varias opciones de configuración que se deben revisar. La primera es crear una cuenta de usuario. Simplemente se debe entrar como root, y ejecutar el programa *adduser*.

## 2.4 Posibles Problemas

Existen altas probabilidades de que surja algún problema cuando se intenta instalar Linux por primera vez. La mayoría de las veces el problema se debe a una simple confusión. Otras veces, sin embargo, puede ser algo más serio. Se debe controlar exhaustivamente lo que se realiza en cada paso, y **nunca aceptar las opciones por defecto que presenta el programa de instalación**, a menos que se tenga la seguridad de que sean las correctas. Los problemas más comunes que se pueden presentar son los que se detallan a

continuación.



### 2.4.1 Problemas con el Arranque

Cuando se intenta arrancar con el floppy de arranque la primera vez, se pueden encontrar algunos problemas:

- Se produce un error en el floppy u otro dispositivo durante el arranque. Puede ser que el disquete esté físicamente dañado, en cuyo caso se tendrá que volver a preparar sobre un nuevo disquete, o bien que los datos fueran mal copiados al mismo, en cuyo caso se debe verificar si se consiguió la imagen del disquete de arranque correctamente.
- El sistema se “cuelga” durante el arranque o después. Después de que el disquete arranca, se debe ver una serie de mensajes del núcleo, indicando qué dispositivos está detectando y configurando. Después de esto, normalmente se verá un prompt de login, que permite iniciar la instalación:
  1. Después del prompt de LILO, el sistema debe cargar el núcleo desde el floppy. Esto puede llevar varios segundos; y puede verse que está sucediendo pues la luz del floppy permanecerá encendida.
  2. Mientras el núcleo arranca, se probarán los dispositivos SCSI. Si el sistema tiene SCSI, el sistema se bloqueará durante unos 15 segundos mientras se prueban esos dispositivos; y sucede normalmente después de verse la línea `lp_init: lp1 exists (0), using polling driver`
  3. Una vez que el núcleo ha terminado de arrancar, se transfiere el control a los ficheros de arranque que hay en el disquete. Finalmente, se verá un prompt de entrada en el sistema, o bien se entrará en un programa de instalación.

- El sistema informa de errores por falta de memoria durante el arranque. Esto tiene que ver con la RAM que se tiene. La solución a este problema es desactivar la opción de disco RAM durante el arranque. Cada versión sigue un procedimiento diferente para hacer esto.
- El sistema muestra un error como “*permission denied*” o “*file not found*” durante el arranque. Esto es señal de que el disquete de instalación está mal. Si se intenta arrancar con el disquete, y éste es correcto, no deberían salir errores de este tipo.
- El sistema informa del error “*VFS: Unable to mount root*” cuando está arrancando. Este error indica que el sistema de ficheros raíz, que se debe encontrar en el disquete de arranque, no está. Puede ser que el disquete esté mal o que no esté arrancando el sistema de forma correcta. Por ejemplo, en muchas distribuciones en CD-ROM, se necesita que tenga el CD en la unidad durante el arranque.

### 2.4.2 Problemas con el Hardware

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

#### Aislando los Problemas de Hardware

Si aparece un problema que se cree que está relacionado con el hardware, la primero que se debe hacer es intentar aislar el problema. Básicamente, se debe quitar todo el hardware que no sea esencial del sistema, y entonces determinar qué dispositivo está causando el problema, de ser posible reinsertando un dispositivo cada vez. Esto quiere decir que se debe quitar todo el hardware excepto la controladora de discos y la tarjeta de video, y por supuesto, el teclado. Los posibles problemas son:

- Conflicto de dirección o de IRQ: todos los dispositivos de la máquina usan una IRQ, o *Línea de Petición de Interrupción*, para decir al sistema que necesitan que el sistema haga algo por ellos. Se puede interpretar una

IRQ como una cuerda de la que el dispositivo tira cuando necesita que el sistema se haga cargo de alguna petición pendiente. Si hay más de un dispositivo tirando de la misma cuerda, el núcleo no podrá determinar a que dispositivo debe atender. Por eso, hay que verificar que todos los dispositivos que tenga instalados estén usando líneas IRQ únicas. Por lo general la IRQ de un dispositivo puede establecerse por puentes en la tarjeta. En algunos casos, el núcleo que proporciona la instalación está preconfigurado para usar una determinada IRQ para un dispositivo concreto.

- Conflictos con los canales DMA, o de *Acceso Directo a Memoria*, direcciones E/S, y direcciones de memoria compartida. Todos estos términos describen mecanismos a través de los cuales el sistema interactúa con los dispositivos hardware. Nuevamente, la manera sencilla de acercarse a estos problemas es deshabilitar temporalmente los dispositivos en conflicto hasta que se tenga tiempo de determinar la causa del problema.

### Problemas para Reconocer Discos Duros o Controladoras

Cuando arranca Linux, se debe visualizar una serie de mensajes en la pantalla. En ellos el kernel está detectando los diversos dispositivos hardware presentes en el sistema. En algún momento, debe aparecer la línea **Partition check:** seguida de una lista de particiones reconocidas, por ejemplo:

```
Partition check:
```

```
hda:  hda1 hda2
```

```
hdb:  hdb1 hdb2 hdb3
```

Si, por cualquier razón, las particiones no son reconocidas, entonces no se podrá acceder a ellas de ningún modo. Hay varias razones que pueden provocar que esto ocurra:

- Disco duro o controladora no soportados: si se está usando una controladora de disco (IDE, SCSI, o cualquier otra) que no esté soportada por Linux, el kernel no reconocerá las particiones a la hora de arrancar.
- Unidad o controladora mal configurada: a pesar de que la controladora esté soportada por Linux, puede no estar configurada correctamente. Se debe consultar la documentación del disco duro y/o controladora para

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

- Geometría del disco duro no reconocida: algunos sistemas, no almacenan la información de geometría del disco duro en la memoria CMOS, donde Linux espera encontrarla. También, ciertas controladoras SCSI necesitan que se les diga dónde encontrar la geometría de la unidad de modo que Linux reconozca el diseño del disco. En general, cuando se arranca el software de instalación, se puede especificar la geometría del disco en el prompt de arranque de LILO con un comando como: `boot: linux hd=<cylinders>,<heads>,<sectors>` donde `<cylinders>`, `<heads>`, y `<sectors>` corresponden al número de cilindros, cabezas y sectores por pista para el disco duro.

### Problemas con Controladoras y Unidades SCSI

Algunos de los problemas más comunes con controladoras y unidades SCSI son:

- Una unidad SCSI se detecta en todos los ID's posibles: la razón, la unidad esta ligada a la misma dirección que la controladora. Es necesario cambiar la configuración de los jumpers de modo que la unidad use una dirección distinta de la de la propia controladora.
- Linux reporta errores de detección, incluso sabiendo que las unidades están libres de errores. Esto puede estar provocado por cables en mal estado, o malos terminadores. Si el bus SCSI no está terminado en ambos extremos, se puede tener errores accediendo a las unidades SCSI.
- Las unidades SCSI dan errores de timeout. Eso se produce generalmente por un conflicto con las IRQ, DMA o direcciones de unidad. Se debe comprobar también que las interrupciones estén correctamente activadas en la controladora.

- Las controladoras SCSI que utilizan BIOS no son detectadas. La detección de controladoras que usan BIOS fallará si la BIOS está desactivada.
- La unidad CD-ROM u otras unidades de información removible no se reconocen a la hora de arrancar. Se debe probar a arrancar con un CD-ROM (o disco) en la unidad. Esto es necesario en algunos dispositivos.

### 2.4.3 Problemas con el Software

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

- El sistema muestra errores como “*Read error*”, “*file not found*” durante la instalación del software. Esto es indicativo de problemas en los disquetes o cintas de instalación. Muchas distribuciones permiten instalar el software desde una partición DOS del disco duro. Esto puede ser más seguro y más rápido que usarlos directamente.
- El sistema da errores tipo “*tar: read error*” o “*gzip: not in gzip format*”. Este problema suele deberse a errores en los ficheros o en los propios discos. En otras palabras, puede que el diskete no contenga errores, pero sí los datos contenidos en él.
- El sistema informa de errores como “*read\_intr: 0x10*” durante los accesos al disco duro. Esto suele deberse a la presencia de bloques con errores en el disco. Sin embargo, si se reciben estos errores al utilizar **mkswap** o **mke2fs**, el sistema puede estar teniendo problemas para acceder al controlador. Puede ser tanto un problema del hardware o una incorrecta especificación de la geometría del disco.
- El sistema da errores como “*file not found*” o “*permission denied*”. Este problema puede suceder si no están disponibles todos los ficheros necesarios en los disquetes de instalación o si hay problemas con los permisos sobre dichos ficheros.





## Capítulo 3

# Entornos Gráficos



### 3.1 KDE

KDE es una colección de programas, tecnologías y documentación que intenta facilitar el uso de los ordenadores a sus usuarios. Es uno de los entornos gráficos más populares de Linux debido a su facilidad de uso en un entorno bonito y agradable.

KDE busca llenar la necesidad de un escritorio de uso fácil para estaciones de trabajo UNIX<sup>®</sup>, similar a los entornos de escritorio que se pueden encontrar en Mac<sup>®</sup> OS o Windows<sup>®</sup> 95/NT. KDE cubre la necesidad de los usuarios de un entorno de trabajo de uso sencillo. Las herramientas utilizadas para alcanzar este objetivo son:

- Mejor comunicación entre aplicaciones.

- Reutilización de componentes.
- Uso generalizado de arrastrar y soltar.
- Aspecto uniforme y estético.
- Administrador de archivos potente y de uso sencillo.
- Navegador web igualmente potente y fácil de usar.
- Sistema de configuración simple y centralizado.
- Sistema de ayuda que proporciona información en cualquier situación.

### 3.1.1 Antecedentes

En Octubre de 1996, el desarrollador alemán de LyX, Matthias Ettrich, inició el desarrollo de KDE con una petición electrónica al foro de USENET. Poco después, un par de desarrolladores comenzaron a planear y a programar partes del nuevo proyecto. Un año más tarde el administrador de ventanas, el administrador de archivos, el emulador de terminal, el sistema de ayuda y la herramienta de configuración del monitor fueron lanzadas para ser probadas, resultando ser relativamente estables.

En Julio de 1998, se lanzó KDE 1.0. Fue la versión estable durante los siguientes seis meses, mientras los desarrolladores continuaron mejorando KDE sin las restricciones de estabilidad. En enero de 1999 sus mejoras se consolidaron y se integraron para producir KDE 1.1, la nueva versión estándar estable.

El desarrollo continuó desde entonces con KDE 2.0, una reescritura casi completa del escritorio, que se lanzó oficialmente el 23 de Octubre de 2000. Las versiones KDE 2.x evolucionaron a través de seis lanzamientos menores durante un año, que aumentaron sus prestaciones y estabilidad.

### 3.1.2 Pantalla

La pantalla de KDE se divide en tres partes fundamentales:

- Panel de KDE: contiene accesos directos a las aplicaciones más empleadas. El equivalente al menú Start de Windows, esto es el menú a través



Figura 3.1: KDE: Escritorios Virtuales.

del cual se pueden ejecutar las aplicaciones. Al seleccionar este elemento se despliega un menú subdivido en distintas categorías. KDE incluye una gran cantidad de utilidades que se integran con el entorno.

- Escritorio: aparece justo encima del panel de KDE. Al igual que en Windows este elemento contiene íconos que permiten acceder a los elementos más comunes como las unidades de disco o la papelera. En la parte superior del escritorio aparece otra barra, en la que aparecerán botones por cada ventana que se cree.
- Panel de ventanas: permite acceder a todas las ventanas que estén abiertas en los distintos escritorios. Al contrario que otros entornos gráficos X Window permite organizar las ventanas en distintos escritorios virtuales. Para cambiar de escritorio virtual se puede eslegir uno de los cuatro botones que aparecen en el panel que se visualiza en la Fig.3.1 de la pág. 41.

### 3.1.3 Administración de Archivos: Kfm

Una de las operaciones más importantes que se pueden realizar con un entorno gráfico es la administración de archivos. Esto incluye investigar el sistema de archivos, buscar, abrir para editarlos, etc. KDE incluye la herramienta *kfm* que permite realizar todas estas operaciones de una forma sencilla.

Al *kfm* se puede acceder, bien seleccionando un enlace a un dispositivo del sistema en el escritorio, bien seleccionando el ícono del directorio personal en el panel o en el menú de KDE: Home directory.

Esto abre una ventana similar a la que se visualiza en la Fig.3.2 de la pág.42 en la cual se muestra el contenido de una determinada carpeta del sistema.

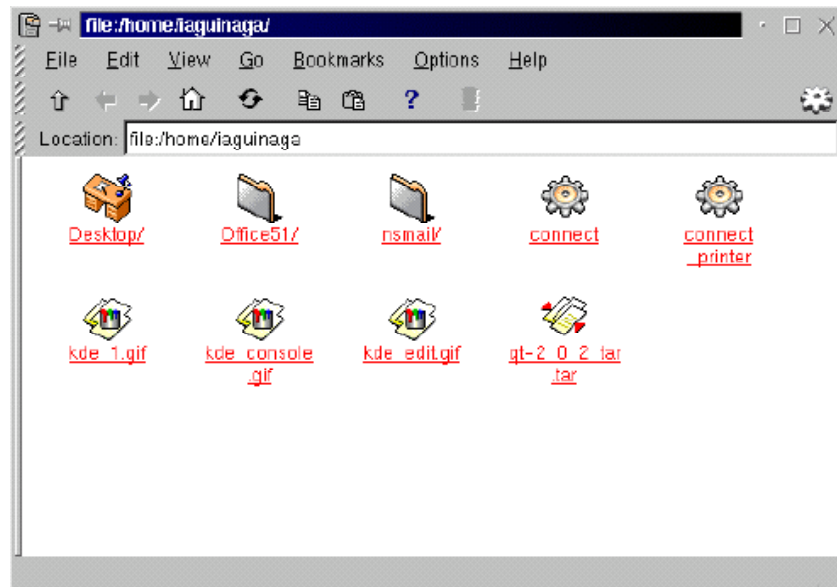


Figura 3.2: KDE: Visualización de Archivos.

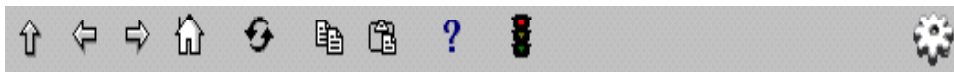


Figura 3.3: KDE: Barra de Navegación.

### 3.1.4 Navegación por la Estructura de Directorios

*Kfm* está diseñado de una manera que su uso y funcionamiento sea muy similar al de un navegador de Internet como Microsoft Internet Explorer o Netscape Navigator. Al igual que estos programas, contiene botones en la barra de herramientas para facilitar la navegación tanto por los directorios como por Internet.

Los botones que aparecen en la barra que se visualiza en la Fig.3.6 de la pág.49, comenzando de izquierda a derecha, realizan las siguientes acciones:

- Sube al directorio padre del actual, pasando por ejemplo de `/home/user1/files` a `/home/user1`, cuando se navega por Internet este botón está

desactivado. También se puede acceder a esta opción desde el menú Go/Up.

- Vuelve al directorio o página Web vista anteriormente a la actual. También se puede acceder a esta opción desde el menú Go/Back.
- Vuelve al directorio o página Web vista posteriormente a la actual. También se puede acceder a esta opción desde el menú Go/Forward.
- El botón Home vuelve al directorio personal del usuario. También se puede acceder a esta opción desde el menú Go/Home.
- El botón Reload redibuja el contenido de la ventana. También se puede acceder a esta opción desde el menú View/Reload Document.
- El botón Copy se emplea para copiar ficheros y directorios. También se puede acceder a esta opción desde el menú Edit/Copy o mediante las teclas <ctrl>+c.
- El botón Paste pega el contenido anteriormente copiado. También se puede acceder a esta opción desde el menú Edit/Paste o mediante las teclas <ctrl>+v.
- El semáforo o botón Stop permite parar la descarga de ficheros de Internet.

### Crear, Copiar, Mover y Borrar Documentos

Una vez ubicado en el directorio en el que se quiere crear un nuevo directorio o carpeta se debe hacer clic con el botón derecho del ratón y en el menú contextual que surge, seleccionar New/Folder o bien, seleccionar el menú File/New/Folder. *Kfm* abre un cuadro de diálogo preguntando por el nombre del nuevo directorio.

El proceso para borrar un directorio o documento es igual de sencillo que el anterior. Para hacerlo no hay más que seleccionar el directorio o documento a borrar y realizar alguna de las siguientes posibilidades según los gustos de cada usuario:

- Hacer clic con el botón derecho del ratón sobre el elemento a borrar y en el menú contextual seleccionar Move to Trash, para mover el fichero

a la papelera (de donde se puede recuperar) o Delete lo cual elimina permanentemente el fichero sin que se pueda recuperar. Se debe tener en cuenta que Linux no dispone de ningún método que permita recuperar un fichero una vez eliminado.

- Seleccionar el elemento y el menú Edit/Move to Trash o presionar las teclas `<ctrl>+x` o bien para eliminar el fichero permanentemente Edit/Delete o las teclas `<ctrl>+supr.`

Para copiar o mover un fichero o directorio (incluyendo su contenido) no hay más que tener dos ventanas con los directorios de origen y destino (A estos efectos el escritorio se comporta como una carpeta más). Para copiar o mover un fichero o directorio no hay más que arrastrar los elementos a la ventana de destino. Tras esto *kfm* mostrará un menú contextual que permite copiar el elemento (Copy), moverlo (Move) o crear un enlace (Link) con lo cual se puede asignar un nuevo nombre o alias al fichero.

### Enlaces de KDE

KDE incluye un tipo de enlace similar al que existe en Windows. Este tipo de enlace se representa por un fichero con extensión `.kdelnk` que contiene información diversa para el uso del sistema. Existen varios tipos de estos enlaces:

- **File System Device:** es un enlace a un dispositivo del sistema. Este tipo de enlaces permite acceder de forma directa a un disquete por ejemplo. Cuando se selecciona New/File System Device en el menú contextual, se abre un primer cuadro de diálogo en el que se pide el nombre del fichero. En el segundo cuadro de diálogo, seleccionando la pestaña Device se pueden indicar las propiedades del enlace como el dispositivo `/dev/fd0` (el disquete) o los íconos del dispositivo cuando está montado y cuando no lo está. Tras esto, al hacer clic sobre el enlace el sistema monta el sistema de archivos del disquete y muestra su contenido. Para sacar el disquete de forma segura hay que seleccionar con el botón derecho el enlace y en el menú contextual seleccionar Umount con lo que se garantiza que ninguna aplicación pueda acceder al disquete y que no haya ninguna que esté accediendo en ese momento. Tras esto se puede retirar el disquete.

- **FTP Url:** es un enlace a un fichero al que se puede acceder a través del protocolo FTP de Internet. El proceso de creación es semejante al anterior indicando el nombre del enlace en el primer cuadro de diálogo y la dirección del enlace del tipo `ftp://servidor/fichero` en la pestaña URL del segundo.
- **Mime Type:** es un enlace que representa una asociación entre un tipo de fichero y las aplicaciones e íconos que lo van a representar. Este proceso permite asociar un tipo de fichero a un programa.
- **Application:** es un enlace a un programa ejecutable. El proceso de creación es el mismo visto anteriormente, primero se indica el nombre del enlace, y luego en la pestaña Execute se puede elegir la aplicación a ejecutar mediante su comando y su ícono.

### Asociar un Nuevo Tipo de Archivo

La asociación de un fichero a otro es un proceso sencillo puesto que lo único que hay que hacer es crear un enlace como se ha visto anteriormente. El primer paso consiste en seleccionar el menú Edit/Mime Types, con lo cual *kfm* abre un directorio en el que se aprecian distintas categorías de ficheros como application, image, text, etc. Para crear una asociación en alguna de estas categorías mencionadas anteriormente se debe crear un enlace de tipo Mime Type, el sistema solicitará que se defina un nombre para el nuevo enlace. En el segundo diálogo se puede definir las propiedades del nuevo tipo de archivo creado.

### Propiedades de un Fichero o Directorio

*Kfm* permite configurar los directorios de forma que se puede seleccionar una imagen de fondo y un ícono distinto del que aparece por defecto. Para ello hay que seleccionar y hacer clic con el botón derecho sobre la carpeta a modificar. En el menú contextual que surge a continuación se selecciona la opción Properties con lo que se muestra un cuadro de diálogo con las propiedades de la carpeta. Haciendo clic en el botón de la carpeta se abre un cuadro de diálogo en el que se permite la selección de cualquier imagen como ícono. Si por el contrario, se modifican las propiedades del fondo se conseguirá que cambie el fondo que muestra *kfm* al abrir la carpeta. Tanto las propiedades de los directorios como de los ficheros ordinarios, que se obtienen de la misma forma,

permiten realizar algunas operaciones básicas sobre los archivos como cambiar el nombre de los mismos y los permisos.

### Configurar *kfm* como Navegador de Internet

*kfm* puede actuar como un navegador de Internet pudiendo visualizar los ficheros HTML. Cuando *kfm* encuentra un directorio en el que existe un fichero llamado **index.htm** muestra de forma automática el contenido del mismo.

Además de mostrar ficheros locales, puede mostrar ficheros a través de Internet, para lo que tiene que ser configurado de una forma muy sencilla y similar a la de otros navegadores como Netscape Navigator, a través del menú Options/Configure Browser. En el cuadro de diálogo que se despliega se debe introducir el proxy que empleará *kfm* para acceder a Internet.

### 3.1.5 Aplicaciones

KDE dispone de una cantidad enorme de aplicaciones auxiliares que permiten realizar las operaciones más habituales de una forma muy sencilla.

#### Konsole

Linux es un sistema Unix y como tal pone a disposición de los usuarios la posibilidad de comunicarse con el sistema a través de una línea de comandos, el **shell**. Desde KDE se puede acceder al shell o consola a través de este programa.

Mediante una ventana como la que se visualiza en la Fig. 3.4 de la pág. 47 se puede configurar el aspecto de la presentación adaptándola a los gustos del usuario, cambiando el esquema de color, las fuentes, el tamaño, entre otras, a través de las distintas opciones del menú Options.

Konsole se integra con el resto de las aplicaciones de KDE mejorando su facilidad de uso. En concreto se pueden arrastrar ficheros y carpetas desde una ventana del administrador de archivos hasta la consola con lo que se permite copiar el path del fichero o cambiar al directorio que contiene un determinado fichero.



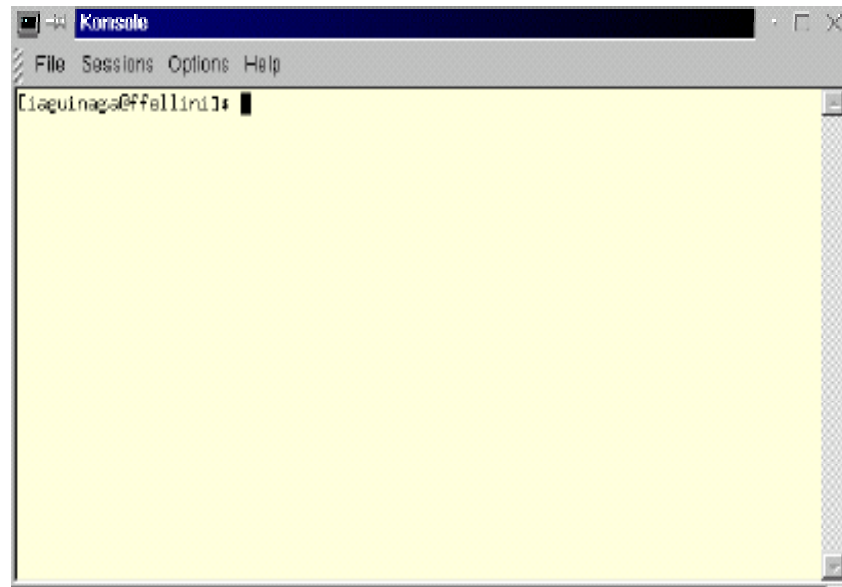


Figura 3.4: KDE: Aplicación Konsole.

### **Kedit**

Kedit es un programa muy sencillo e intuitivo para realizar la edición de textos sencillos. El manejo de kedit es similar al de programas como Notepad, al que se accede a través del menú Application/Text Editor. Admite las opciones típicas de manejo de textos como son copiar, pegar, cortar; además de otras más sofisticadas como insertar un fichero, una fecha, buscar un texto en el documento, reemplazar texto, o comprobar la ortografía del documento. El programa es además muy configurable, puesto que permite definir el idioma del texto (Options/Spellchecker) o la fuente con la que se va a mostrar.

### **Kwrite**

Kwrite, al igual que kedit, es un programa especializado en la manipulación de ficheros de texto, pero a diferencia de este está orientado al desarrollo de programas por lo que ofrece la posibilidad de colorear la sintaxis de los mismos empleando distintos lenguajes de programación: C, C++, Java, HTML, Bash, Modula 2, Ada, Python o Perl.

## Kdehelp

Esta aplicación representa el sistema de ayuda del sistema. Se basa en HTML por lo que su uso es muy sencillo y similar al de un navegador de Internet. Todas las aplicaciones del KDE acceden a este programa para mostrar la ayuda de los mismos. La ventana de kdehelp se divide en cuatro partes fundamentales: la barra de menús, la barra de herramientas, la barra de direcciones, y el contenido propiamente dicho.

Esta ayuda de KDE está llena de vínculos que llevan de un contenido a otro. Para navegar por los documentos existen las opciones típicas de todos los navegadores y que se encuentran también en kfm.

Una de las opciones más interesantes de kdehelp es que permite el acceso a las páginas del manual man de Linux, simplemente escribiendo `man:<comando>` donde comando es alguno de los comandos de Linux.

## Kfind

Esta herramienta es auxiliar a kfm puesto que permite buscar un determinado archivo en un directorio concreto. La búsqueda al igual que en Windows, se puede realizar siguiendo tres criterios diferentes, como la muestra la Fig. 3.5 de la pág. 49.

- Por nombre u localización, se conoce el nombre o parte de él y la localización aproximada del fichero.
- Se puede centrar más la búsqueda si se conoce el momento en el que se realizó la última modificación. La pestaña Date Modified permite que el usuario identifique un período de tiempo en el que concretar la búsqueda.
- Se puede especificar en Advanced que la búsqueda se limite a un determinado tipo de fichero, que el fichero contenga un determinado texto o tamaño de letra.

Una vez determinados los criterios de selección de los ficheros se debe indicar al programa que busque seleccionando el primer botón de la barra de herramientas por la izquierda, o el menú File/Start Search., con lo que el programa comenzará a buscar.

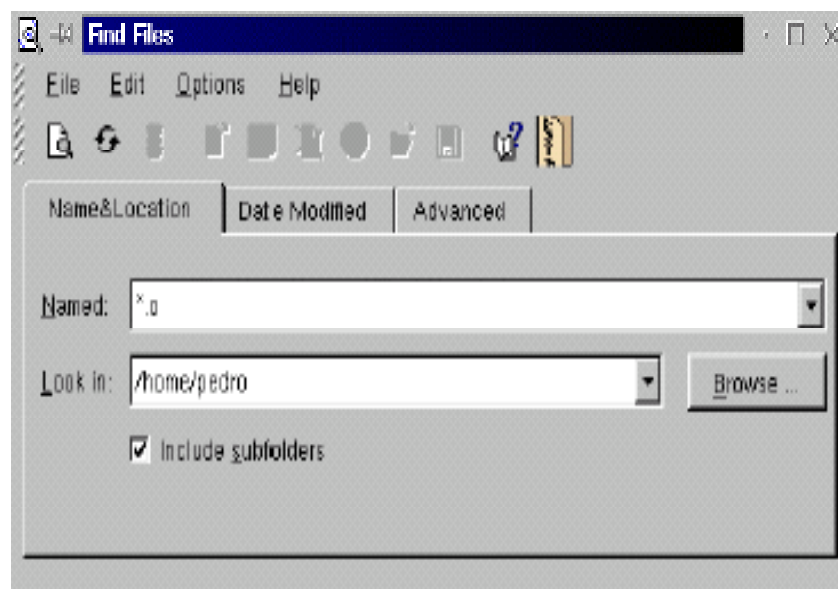


Figura 3.5: KDE: Aplicación Kfind.



Figura 3.6: KDE: Barra de Herramientas de Kfind.

La barra de herramientas de esta aplicación, que se visualiza en la Fig. 3.6 de la pág. 49 contiene las siguientes funciones:

1. Botón para iniciar una búsqueda.
2. Botón para crear una nueva búsqueda.
3. Botón para detener búsqueda (desactivado en la imagen).
4. Botón para abrir fichero.
5. Botón para añadir a un archivo tar en el que se pueden agrupar varios ficheros.
6. Botón para eliminar fichero.
7. Botón de propiedades.
8. Botón para abrir el directorio que contiene el fichero.
9. Botón salvar resultados.
10. Botón de guardar resultados.
11. Salir.

### 3.1.6 Configuración

Como cualquier aplicación de Linux que se precie KDE es altamente configurable, lo que supone que cada usuario puede adaptar el aspecto y comportamiento de KDE a su gusto personal. No obstante al contrario que otras muchas aplicaciones para Linux, para configurar KDE no es necesario editar los ficheros de configuración a mano sino que existen una serie de herramientas gráficas que permiten estos cambios de una forma sencilla y segura.

#### Editor de Menús

Uno de los aspectos más sencillos de cambiar es el menú de aplicaciones del sistema al que se pueden añadir las aplicaciones de uso más común. Existe con este fin una utilidad llamada editor de menús accesible desde el menú

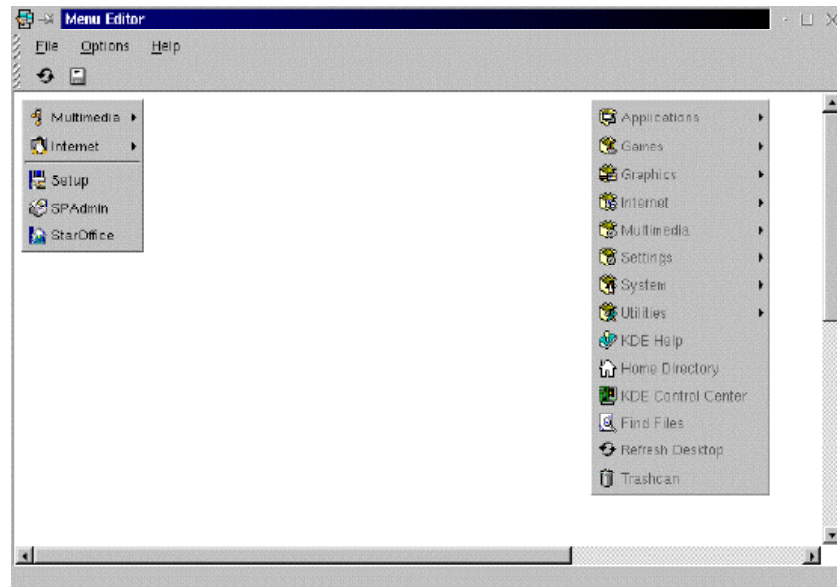


Figura 3.7: KDE: Menú Editor.

Utilities/Menu Editor. Una vez arrancada la aplicación aparecerá una imagen semejante a la que se aprecia en la Fig. 3.7 de la pág. 51.

El menú de KDE está compuesta por dos partes principales. La de la izquierda constituye el menú personal del usuario en el que puede añadir o quitar aplicaciones. La de la derecha es común a todos los usuarios de KDE y por lo tanto sólo puede ser modificada por el administrador del sistema.

En cualquiera de los casos el proceso para crear una nueva entrada en el menú es muy sencilla. Se pulsa con el botón derecho del ratón sobre el menú (o submenú) que se vaya a modificar, con lo que se despliega un menú contextual con diversas opciones:

- *Change*: permite editar las propiedades de la entrada del menú sobre la que se haya hecho la selección, editando su nombre, la aplicación que arranca, el icono, etc.
- *Select item for moving*: permite cambiar la posición del un elemento del menú, para lo cual hay que hacer clic en el menú y arrastrar el elemento a su nueva posición.

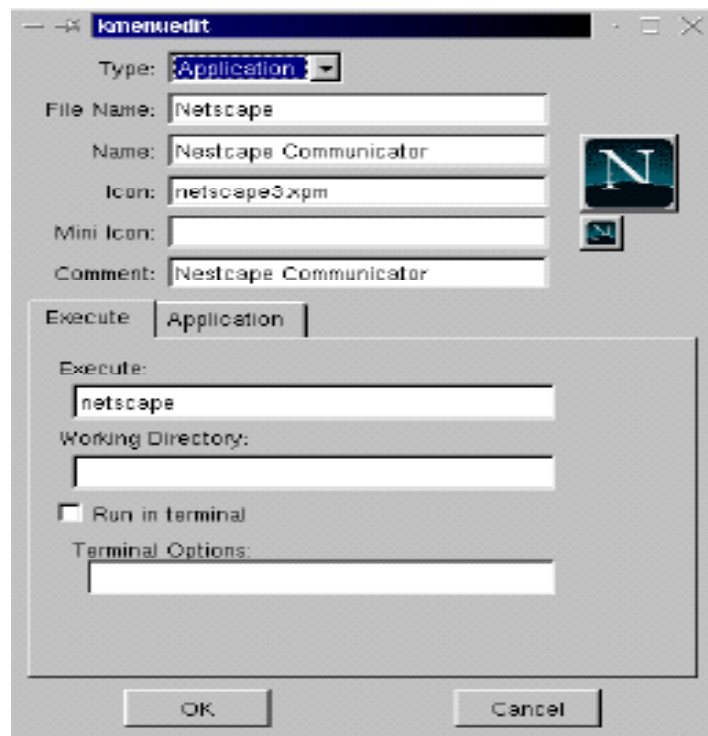


Figura 3.8: KDE: Cuadro de Diálogo.

- *Select menu for moving*: igual que el anterior pero con menús completos.
- *New*: se crea un nuevo elemento del menú.
- *Cut*: se corta un elemento del menú.
- *Copy*: se copia un elemento del menú.
- *Paste*: se pega un elemento previamente cortado o copiado.
- *Delete*: se elimina el elemento del menú seleccionado.

Tanto si se modifican las propiedades de una entrada del menú existente o se crea un nuevo elemento el programa presenta un cuadro de diálogo, como el que se visualiza en la Fig. 3.8 de la pág. 52.

En el cuadro de diálogo se pueden definir los siguientes aspectos:

- *Type*: tipo del elemento creado puede ser Separator, Submenu, Application, Swallow, Link, o Device, dependiendo del tipo que se escoja la parte inferior del cuadro de diálogo cambiará permitiendo configurar cada uno de los tipos.
- *File Name*: el nombre del fichero en el que se va a guardar la información del menú.
- *Name*: el nombre que aparecerá en el menú una vez creado el mismo.
- *Icon*: el nombre de la imagen que aparecerá en el menú.
- *Mini Icon*: la imagen que aparecerá cuando sea necesario mostrar un ícono pequeño. Si se deja esta opción en blanco KDE mostrará una versión reducida de la imagen que aparezca en Icon.
- *Comment*: un comentario que pueda ayudar a determinar que hace esa entrada del menú.

### KDE Control Center

Esta aplicación es la principal encargada de configurar KDE y a ella se puede acceder de muchas formas, tanto desde el ícono que aparece en el panel, como desde cualquiera de las entradas al menú Settings, en cuyo caso sólo se accede a una de las posibles opciones de configuración. Cuando se arranca la figura a aparece una ventana similar a la que se visualiza en la Fig. 3.9 de la pág. 54.

En la parte de la izquierda aparecen ordenadas las diferentes categorías de configuración (que coinciden con las categorías del menú Settings), mientras que en la derecha se abren los distintos cuadros de diálogo que permiten configurar KDE.

Otra de las tareas de facilitan el uso de KDE es la posibilidad de añadir una aplicación al panel de forma que sea fácilmente accesible. El proceso a seguir es simplemente elegir un elemento del menú a través de Panel/Add Application, con lo que se despliega un menú idéntico al de KDE con las aplicaciones.

#### 3.1.7 Otras Aplicaciones

Algunas aplicaciones auxiliares son:



Figura 3.9: KDE: Control Center.

- Korganizer: es un programa que permite gestionar la agenda del usuario de una forma sencilla e incluso la sincronización de datos con agendas personales.
- Icon Editor: permite crear iconos para personalizar los menús y enlaces de KDE.
- Kview: permite mostrar imágenes de todos los formatos de archivo importantes así como realizar operaciones sencillas con ellas.
- Paint: programa de dibujo básico que permite crear imágenes sencillas.
- PS Viewer: permite visualizar ficheros con imágenes PostScript y documentos de Adobe Acrobat (.pdf).
- SnapShot: permite la captura de una ventana y su contenido.
- Desktop Switching Tool: aplicación muy útil puesto que permite seleccionar cual va a ser el entorno por defecto que arranque Linux, permite seleccionar entre GNOME, KDE y AfterStep.
- Archiver: permite el manejo de ficheros tar y zip de una forma sencilla.





Figura 3.10: GNOME: Pantalla de Inicio.

## 3.2 GNOME

Cuando Linux arranca en modo gráfico aparece una ventana similar a la que se visualiza en la Fig. 3.10 de la pág. 55.

En ella se puede introducir el nombre del usuario y la clave.

Cuando se sale del sistema vuelve a aparecer esta misma ventana. Para apagar el ordenador se puede seleccionar el botón de *Options* tras lo que aparece un menú con las siguientes opciones:

- Sessions, permite elegir al usuario el entorno de ventanas con el que va a trabajar, los entornos más habituales son, AnotherLevel, Default (arranca el entorno por defecto instalado que puede ser cualquiera de los otros), FailSafe (Modo a prueba de fallos), Gnome o KDE.
- Language, permite cambiar el idioma en el que se muestran algunos de los mensajes del sistema.
- System, contiene dos opciones, Reboot (para rearrancar el sistema) y Halt para apagarlo.



- Applets: son enlaces a las aplicaciones de uso más frecuente como la consola, Netscape, ayuda, etc.
- Acceso a los escritorios virtuales. Al contrario que en Windows, X Window permite organizar las ventanas en varios escritorios virtuales.

Al hacer click con el botón derecho del ratón sobre cualquiera de los elementos anteriores aparecerá un menú contextual que permite configurar el elemento. Las cuatro opciones más habituales son:

- *Remove from Panel*: con lo que se elimina el applet del panel.
- *Move applet*: permite modificar la posición del applet arrastrándola y soltándola en la nueva posición.
- *About*: muestra información sobre el autor del applet.
- *Properties*: abre un cuadro de diálogo en el que se permite personalizar todos las características del applet.

De igual forma que en el panel, también se puede hacer click con el botón derecho en el escritorio, lo que despliega un menú contextual con diversas opciones.

### 3.2.2 Aplicaciones

A continuación se comentan brevemente las aplicaciones que por defecto incluye el entorno GNOME.

#### File Manager

Al crear una cuenta de usuario el sistema crea un directorio o carpeta personal en la que el usuario puede crear y modificar los ficheros con los que trabaja. GNOME incluye una herramienta, que permite la manipulación de los ficheros.

El administrador de archivos permite seleccionar ficheros y/o directorios y moverlos, copiarlos o eliminarlos. Para seleccionar un único fichero o un directorio basta con hacer clic sobre él. Para seleccionar varios archivos y/o directorios basta con hacer clic sobre ellos manteniendo pulsada la tecla <Ctrl>.

Si están contiguos también se pueden seleccionar manteniendo pulsada la tecla <Shift> y haciendo click en los ficheros de los extremos.

Para mover un directorio, un archivo o una selección basta con mantener clickeado el ratón sobre él y arrastrarlo a su nueva posición. Para copiar un directorio, un fichero o una selección se realiza la misma operación pero manteniendo pulsada la tecla <Ctrl>. Finalmente para eliminar directorios, ficheros o una selección hay que hacer click sobre ellos con el botón derecho del ratón hasta que aparezca un menú en el que se debe escoger la opción *delete*. Si se realiza un doble click sobre un fichero se abre un cuadro de diálogo que pregunta el programa con el que se desea abrir ese fichero. Si se ha realizado un doble click sobre un directorio se mostrarán los ficheros y subdirectorios que contiene. En el caso de haberse realizado un doble click sobre un archivo comprimido se mostrarán, igual que si se tratase de un directorio, los ficheros y subdirectorios que contiene (no es necesario recurrir a otros programas para manipular su contenido).

En el menú **Edit/Preferences/File Display** se puede elegir entre ver también los archivos ocultos o no (los archivos ocultos son aquellos cuyo nombre empieza con un punto). El ícono **Rescan** permite actualizar la lista de ficheros tras realizar un cambio.

En el caso de no saber exactamente donde se encuentra un fichero se puede resultar útil la opción **Find File** del menú **Commands**. Se debe indicar el directorio donde se empezara a realizar la búsqueda (hay que recordar que el punto representa el directorio actual) y el nombre del fichero buscado. Si se conoce parte del nombre se pueden utilizar los caracteres comodines ? y \*.

## Search Tool

Este programa se encuentra en el menú **Utilities** y permite buscar ficheros que cumplan una serie de criterios empezando a buscar en el directorio que se le indique. Los criterios de búsqueda los puede activar y desactivar el usuario con el botón *Enable* o incluso quitarlos de la ventana mediante el botón *Remove*. Para añadir criterios se selecciona el criterio deseado y se añade mediante el botón *Add*.

Una vez de que el usuario haya editado los criterios seleccionados deberá pulsar el botón *Start* y se mostrarán los ficheros encontrados que cumplen los criterios indicados por el usuario.

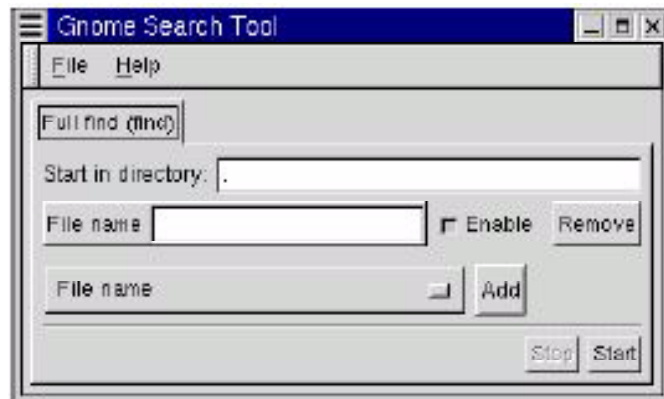


Figura 3.13: GNOME: Aplicación Search Tool.

### Color Xterm, GNOME Terminal y Regular Xterm

Se encuentran en el menú **Utilities** y sirven para abrir una consola de Linux en la cual introducir comandos.

### Multimedia

El menú Multimedia incluye diversas aplicaciones:

- Audio Mixer: controla el volumen de las diversas fuentes de sonido (micrófono, CD-ROM, Speaker y Line). Permite ajustar otra serie de parámetros como ser la amplificación. También da la opción de silenciar las fuentes de sonido que se desee.
- CD Player: permite la reproducción de CDs de música.

### 3.2.3 Configuración

Como cualquier aplicación de Linux GNOME es altamente configurable, para su configuración existen varias herramientas. La herramienta principal es **GNOME Control Center**. Se puede acceder a ella mediante el menú **Settings / GNOME Control Center** o mediante el ícono del panel con aspecto de caja de herramientas.

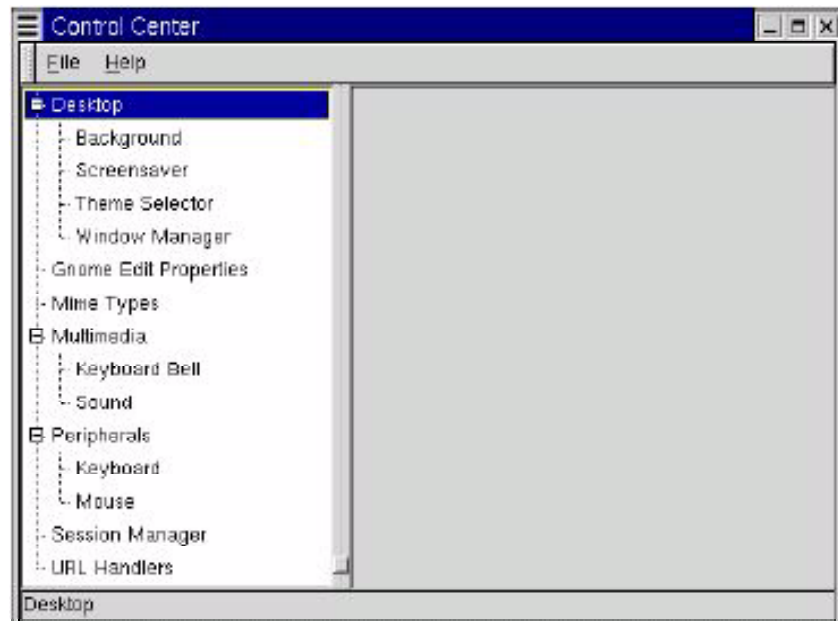


Figura 3.14: GNOME Control Center.

Esta aplicación, tal como lo visualiza la Fig. 3.14 de la pág. 60, permite configurar muchos aspectos de la apariencia de GNOME, como el fondo de pantalla o los sonidos que emplearán los programas. Todos los cambios que se realicen afectarán únicamente al usuario que los ha realizado.

En el apartado **Background** el usuario puede elegir el fondo de pantalla que desee para sus escritorio. Este fondo puede ser una foto o un mosaico de un determinado patrón o un gradiente de colores.

En el apartado **ScreenSaver** el usuario puede elegir el salvapantallas que desea utilizar. Un salvapantallas no es más que un programa gráfico que se activa cuando la pantalla lleva mucho tiempo sin cambiar. Al activarse el salvapantallas el contenido de la pantalla cambia y evita que la pantalla se desgaste prematuramente por mostrar siempre la misma imagen estática.

Una característica interesante del salvapantallas es la opción **require password**. Al activar esa opción el usuario deberá introducir la contraseña para desactivar el salvapantallas. De esa manera usuario se asegura que nadie accederá a sus archivos mientras esté ausente.

El apartado **Theme Selector** permite al usuario elegir entre distintas combinaciones de colores y fuentes para realizar la representación de los elementos, como botones y menús de las ventanas.

En el apartado **Edit Properties** el usuario puede elegir el editor que se usará por defecto para abrir o editar ficheros de texto.

El apartado **Mime Types** permite al usuario definir los programas que se usarán por defecto para abrir, visualizar o editar un cierto tipo de archivos. Además permite añadir nuevos tipos de archivos.

En el apartado **Multimedia / Keyboard Bell** el usuario puede elegir el volumen, el tono y la duración que utilizará el altavoz del ordenador. Mientras que en el apartado **Multimedia / Sound** el usuario puede activar o desactivar los efectos sonoros. Además puede elegir el sonido que desea asociar a cada evento del sistema.

Entrando en el apartado **Peripherals/Keyboard** el usuario puede modificar la rapidez de repetición de caracteres cuando se mantiene pulsada una tecla y el retraso temporal desde que se pulsa una tecla hasta que se empiezan a repetir los caracteres si se mantiene pulsada. En el apartado **Peripherals / Mouse** el usuario puede personalizar la rapidez de respuesta del ratón así como indicar su uso por parte de una persona diestra o una persona zurda.

En **Session Manager** el usuario puede indicarle al sistema que le pida confirmación antes de acabar la sesión, es decir, antes de hacer logout. También puede indicarle al sistema que guarde la configuración de GNOME automáticamente o no. La última opción de este apartado es la de elegir que programas se arrancarán automáticamente cada vez que el usuario comience una nueva sesión.

El último apartado de este programa es el apartado **URL handler**. En este el usuario puede seleccionar que programas se utilizan para mostrar URLs.

En el apartado **Window Manager** el usuario tiene la opción de elegir y configurar el gestor de ventanas. La herramienta de configuración del gestor de ventanas de Enlightenment es similar al que se visualiza en la Fig. 3.15 de la pág. 62.

En el apartado **Basic Options** el usuario puede seleccionar el modo de arrastre y de cambio de tamaño de las ventanas. También permite indicar al sistema que la ventana activa sea siempre aquella sobre la que se encuentra el

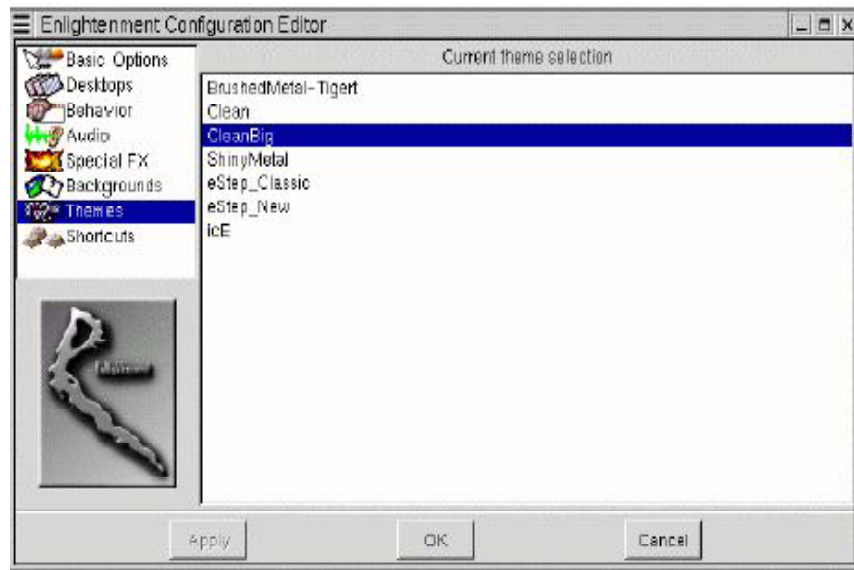


Figura 3.15: GNOME: Enlightenment Configuration Editor.

cursor o la última ventana sobre la que se ha hecho click.

En el apartado **Desktops** el usuario puede elegir el número de pantallas virtuales con las que se desea trabajar (por defecto 4). Además por cada pantalla virtual se pueden crear más pantallas mediante *Separate Desktops*. **Edge Flip Resistance** indica el intervalo de tiempo que debe transcurrir para pasar al Desktop virtual contiguo cuando el ratón se encuentre en un borde de la pantalla. Si está desactivado para pasar de un Desktop a otro habrá que utilizar el **Applet Gnome Pager** que se encuentra en el panel de GNOME.

En el apartado **Behaviours** el usuario puede modificar el comportamiento de las ventanas. El programa de configuración permite elegir cuándo y cómo se desea que una ventana se active así como configurar los mensajes de ayuda que aparecen cuando se deja el ratón quieto sobre algún elemento de un programa.

En el apartado **Sounds** el usuario tiene la única opción de activar o desactivar los efectos de sonido del gestor de ventanas.

El apartado **Background** permite elegir el fondo del escritorio. El fondo puede ser simplemente un color sólido seleccionado por el usuario o una ima-



gen también seleccionada por el usuario. Se puede crear una lista de fondos preferidos.

En el apartado **Themes** el usuario puede elegir entre distintas combinaciones de colores y fuentes para realizar la representación de los elementos, como botones y menús de las ventanas.

En el apartado **Shortcuts** el usuario puede editar las operaciones que se realizan pulsando una combinación de teclas concreta.

Hay varias formas de configurar el panel de GNOME. Clickeando sobre cualquier elemento de la barra del panel con el botón derecho del ratón aparece un menú contextual, con la opciones de eliminar dicho elemento del panel de GNOME, cambiarlo de sitio o editar sus propiedades. Si se hace click sobre una zona del panel que este libre aparecerá un menú contextual.

Las tres primeras opciones permiten acceder a varios submenús del menú principal. Mediante el menú **Add applet** se puede añadir al panel de GNOME cualquiera de las distintas applets disponibles clasificadas en 5 grupos:

- *Amusements*: applets de entretenimiento lo que incluye juegos sencillos, animaciones, etc.
- *Monitors*: applets que muestran el estado de diversos dispositivos del sistema como las baterías de un móvil, la carga de trabajo de la CPU, el uso de los discos o memoria, etc.
- *Multimedia*: applets para el manejo de CD-Audio..
- *Network*: applets que muestran el estado de la red y la recepción de mail.
- *Utility*: muestra distintos relojes, caracteres especiales para pegar en los documentos, acceso a escritorios virtuales, entre otros.

Al hacer click con el botón derecho del ratón sobre una zona libre del panel de GNOME aparece el menú contextual que se visualiza en la Fig. 3.16 de la pág. 64.

Con la opción **Add new panel** el usuario puede añadir más paneles. El usuario podrá elegir entre dos tipos de panel: *Corner Panel* y *Edge Panel*. En el primer caso el panel ocupa justo el espacio que ocupen los Applets mientras



Figura 3.16: GNOME: Menú Contextual.

que en el segundo caso el panel ocupa todo el espacio posible yendo de extremo a extremo de la pantalla.

Con la opción **Convert to corner panel** o **Convert to edge panel** el usuario puede cambiar el tipo de panel una vez creado.

Con la opción **Add main menu** el usuario puede añadir el menú principal al Panel.

Con la opción **Add log out button** el usuario puede añadir al Panel un botón que dé por terminada la sesión de trabajo del usuario.

Con la opción **Add new launcher** el usuario puede añadir al panel de herramientas un botón que al ser pulsado ejecute el programa que el usuario haya determinado.

Con la opción **Add Drawer** el usuario puede añadir un “cajón” al panel.

Con la opción **This panel properties** del menú de la página anterior se puede colocar el panel horizontalmente o verticalmente en cualquiera de los cuatro bordes de la pantalla . Si el panel fuese de tipo *corner panel* el

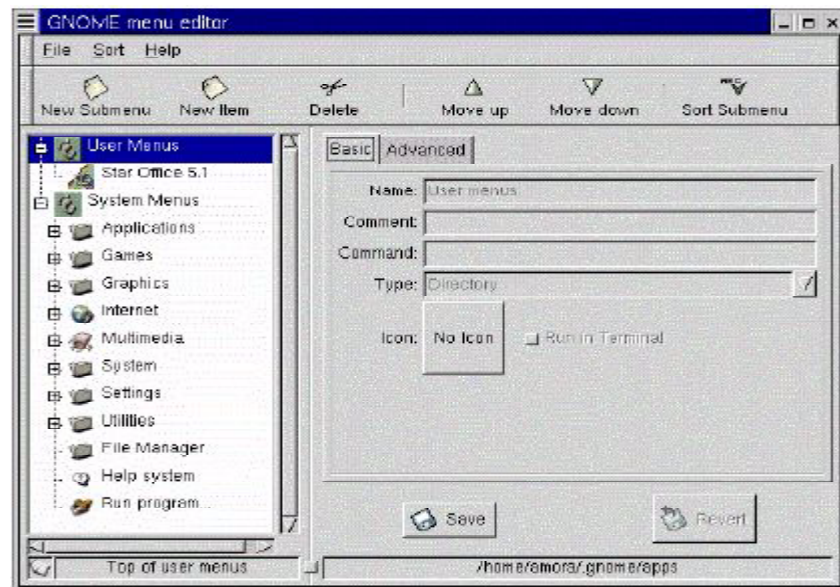


Figura 3.17: GNOME: Menú Editor.

usuario también podría elegir la esquina a la que quiere que se acerque más el panel. Otra opción que permite configurar **This panel properties** es el fondo del panel pudiendo elegirse como fondo el fondo por defecto, un color o una imagen.

Para configurar el menú principal hay que utilizar el programa **Menú Editor** que se encuentra en el menú **/Settings / Menu Editor** del menú principal.

Mediante los iconos *New Submenu* y *New Item* que se visualizan en la Fig. 3.17 de la pág. 65, el usuario puede añadir nuevos submenús o nuevos elementos a los dos menús existentes (User menu y System menu). Para modificar el *System menu* se deberá haber entrado en el sistema como superusuario pero para modificar el *User menu* no es necesario.

### 3.2.4 Otras Aplicaciones

Otras aplicaciones son:

- **GQ View:** se encuentra en el menú **Graphics** y permite visualizar los ficheros gráficos. Permite recorrer directorios y muestra en el lado derecho el gráfico contenido en el fichero seleccionado. Soporta muchos tipos de ficheros gráficos tales como jpg, png, gif, bmp, tiff. No tiene ninguna herramienta de edición (cuando se desea editar la imagen el propio programa arranca aquellos que el usuario ha elegido como editores).
- **GNOME DiskFree:** se encuentra en el menú **Utilities** e indica el espacio ocupado de los dispositivos que están siendo utilizados, es decir, los que están montados.
- **Simple Calculator:** se encuentra en el menú **Utilities** y permite realizar operaciones de cálculo sencillas. En caso de producirse algún error mostrará el símbolo *e*.
- **GnomeCard:** este programa se encuentra en el menú **Applications**. Sirve para gestionar direcciones y para crear tarjetas.
- **Calendar:** se encuentra en el menú **Applications** y es el equivalente electrónico de una agenda. Permite ver el calendario anual y el calendario del mes. También permite ver la agenda semanal y la agenda diaria. En la agenda diaria el usuario puede introducir todas las citas del día. Las citas aparecerán resumidas en la agenda semanal.
- **System Info:** se encuentra en el menú **Utilities** y da información sobre el sistema que se está utilizando. Si se hace click en **Detailed Information** suministra información extra acerca de la CPU, la memoria y el disco o discos duros del ordenador.
- **Change Password:** se encuentra en el menú **System** y su finalidad es más que obvia. Tal como se visualiza en la Fig. 3.18 de la pág. 67, permite al usuario cambiar la contraseña que utiliza para entrar en el sistema. El usuario deberá introducir en primer lugar su contraseña actual y luego la nueva contraseña.
- **System Monitor:** se encuentra en el menú **Utilities**. Como se visualiza en la Fig. 3.19 de la pág. 68, permite monitorizar los sistemas de archivos. También permite monitorizar la memoria utilizada por diversos programas o procesos que se están ejecutando en ese instante. Además de la memoria utilizada por cada proceso, indica el tanto por ciento de CPU que utiliza y el propietario del proceso. Si se hace click dos veces



Figura 3.18: GNOME: Aplicación Change Password.

sobre cualquiera de los procesos se muestra información detallada de ese proceso mientras que si se hace click con el botón derecho del ratón el usuario puede decidir entre ver los detalles del proceso, reiniciarlo, o enviar diversas señales. Si alguno de los procesos se ha quedado bloqueado se puede arrancar este programa y eliminar el proceso en cuestión enviándole una señal Kill o Term.

- Gedit: se encuentra en el menú **Applications**. Se trata de un sencillo editor de textos gráfico. Es útil para editar diversos ficheros de configuración. Soporta la opción de copiar y pegar texto así como la capacidad de búsqueda de palabras en el documento. En el menú **Settings** hay ciertas opciones de configuración como ajustar las líneas al tamaño de la pantalla. En el menú **Settings / Preferences** se puede elegir el tipo de fuente y los elementos que el usuario desea utilizar en el menú **Plugins**. Entre los plugins que incluye por defecto el programa destacan el corrector ortográfico, y el plugining Email. Este último plugining permite enviar directamente como correo electrónico el archivo que actualmente se está editando sin necesidad de abrir ningún otro programa.
- Time tracking tool: esta herramienta se encuentra en el menú **Applications**. Esta diseñada para medir el tiempo que utiliza el usuario en realizar alguna tarea. Esta utilidad permite al usuario conocer cuáles son los programas con los que pasa más tiempo. El funcionamiento de esta herramienta es *manual*, el usuario debe crear un proyecto en *GtimeTracker* y cuando vaya a utilizar el programa en cuestión deberá activar el proyecto para que el tiempo empiece a contar.
- Gnumeric spreadsheet: este programa se encuentra en el menú **Applications**. Se trata de una hoja de cálculo. En cada celda el usuario puede introducir texto, fechas, valores numéricos o fórmulas matemáticas.

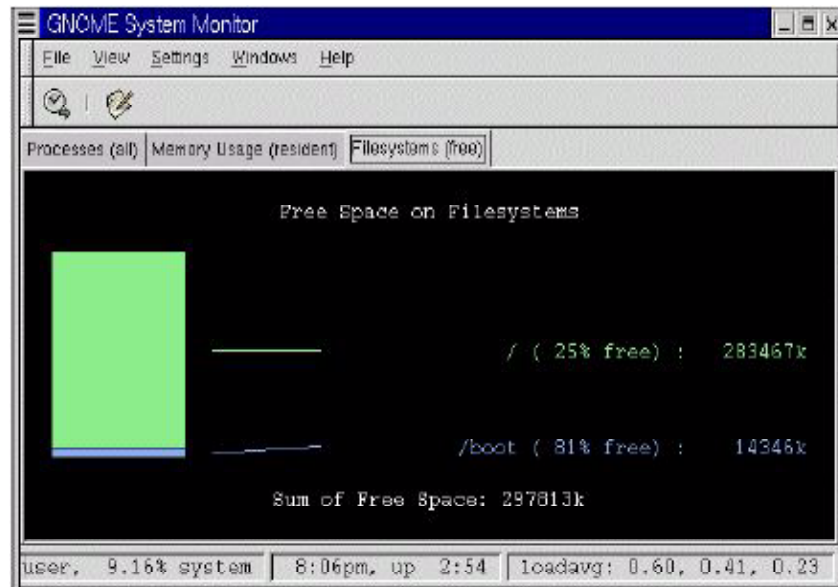
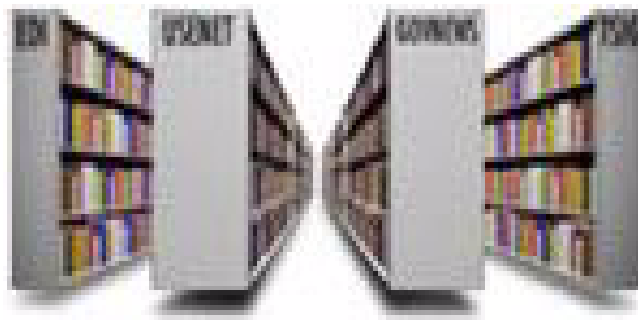


Figura 3.19: GNOME: Aplicación System Monitor.

En las fórmulas matemáticas intervienen celdas con valores numéricos previamente rellenados y funciones que aporta el programa tales como SUM, AVERAGE, entre otras.

## Capítulo 4

# Sistema de Ficheros



### 4.1 Introducción

El sistema de archivo de Linux sigue todas las convenciones de Unix, lo cual significa que tiene una estructura determinada, compatible y homogénea con el resto de los sistemas Unix. Al contrario que en Windows o MS-DOS el sistema de archivos en cualquier sistema Unix no está ligado de una forma directa con la estructura del hardware, esto es, no depende de si un determinado ordenador tiene 1, 2 o 7 discos duros para crear las unidades `c:\`, `d:\` o `m:\`.

Todo el sistema de archivos de Unix tiene un origen único la raíz o root representada por `/`. Bajo este directorio se encuentran todos los ficheros a los que puede acceder el sistema operativo. Estos ficheros se organizan en distintos directorios cuya misión y nombre son estándar para todos los sistemas Unix.

No existe, a nivel de usuario, el concepto de volumen, ni de dispositivo físico. Es decir, el usuario no sabe en qué disco están los ficheros que está utilizando.

El sistema de ficheros es la estructura que permite que Linux maneje los archivos que contiene.

Todos los archivos de Linux tienen un nombre, el cual debe cumplir con ciertas reglas:

- Un nombre de archivo puede tener entre 1 y 255 caracteres.
- Se puede utilizar cualquier carácter excepto la barra inclinada / y no es recomendable emplear los caracteres con significado especial en Linux, que son los siguientes: `\ ^ ~ ' " ' * ; - ? [ ] ( ) ! & ~ < >`. Para emplear ficheros con estos caracteres o espacios hay que introducir el nombre del fichero entre comillas.
- Se pueden utilizar números exclusivamente si así se desea. Las letras mayúsculas y minúsculas se consideran diferentes.

Como en Windows, se puede emplear un cierto criterio de “tipo” para marcar las distintas clases de ficheros empleando una serie de caracteres al final del nombre que indiquen el tipo de fichero del que se trata. Así, los ficheros de texto, HTML, las imágenes PNG o JPEG tienen extensiones `.txt`, `.html`, `.png` y `.jpg` (o `.jpeg`) respectivamente.

La extensión es opcional. En caso de que exista, por convenio significa:

- **.f** Programa fuente escrito en lenguaje FORTRAN.
- **.p** Programa fuente escrito en lenguaje PASCAL.
- **.c** Programa fuente escrito en lenguaje C.
- **.o** Fichero objeto.
- **.a** Biblioteca de módulos.
- **.h** Fichero de “cabecera”.

Pese a esto Linux sólo distingue tres tipos de archivos:



- Archivos o ficheros ordinarios, son los mencionados anteriormente.
- Directorios (o carpetas), son archivos especiales que agrupan otros ficheros de una forma estructurada.
- Archivos especiales, que están asociados a dispositivos de entrada/salida. Contienen referencias a los drivers (programas que manejan directamente los dispositivos y que forman parte del núcleo). Pueden ser de tipo “*bloque*” (apuntan a dispositivos tipo disco) y “*carácter*” (apuntan a dispositivos como terminales, impresoras, etc). Por convenio, residen en el directorio `/dev`.

## 4.2 Enlaces

Los enlaces son un tipo de archivo ordinario cuyo objetivo es crear un nuevo nombre para un archivo determinado. Una vez creado el enlace simbólico éste permite acceder al fichero que enlaza de igual modo que si se hubiera copiado el contenido del mismo a otro fichero, con la ventaja de que este realmente no se ha copiado. Los enlaces simbólicos son especialmente útiles cuando se quiere que un grupo de personas trabaje sobre un mismo fichero, puesto que permite compartir el fichero pero centraliza las modificaciones.

## 4.3 El Camino o Path

En cualquier sistema operativo moderno la estructura de archivos es jerárquica y depende de los directorios. En general la estructura del sistema de archivos se asemeja a una estructura de árbol, similar a la que se visualiza en la Fig. 4.1 de la pág. 72, estando compuesto cada nudo por un directorio o carpeta, que contiene otros directorios o archivos.

En los sistemas Unix, y por lo tanto en Linux, existe una única raíz simbolizada con `/` de la que cuelgan todos los ficheros y directorios, y que es independiente de qué dispositivos estén conectados al ordenador.

La referencia a un fichero puede ser en referencia al directorio actual (vía de acceso relativa) o en referencia al directorio raíz (vía de acceso absoluta). En la vía de acceso relativa se puede utilizar la notación `..` que significa una referencia al directorio padre.

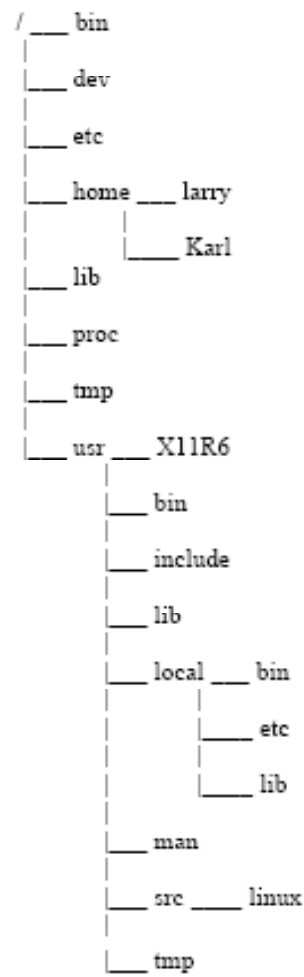


Figura 4.1: Árbol de Directorios.

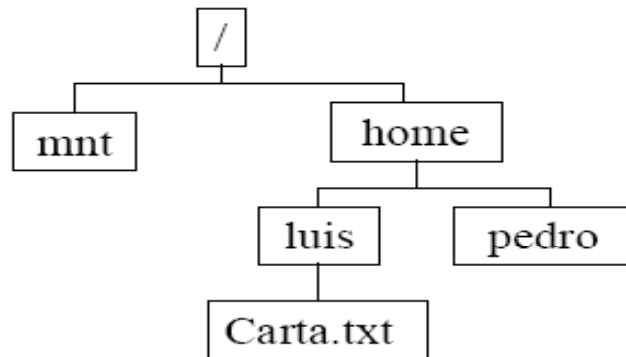


Figura 4.2: Ejemplo de Estructura de Ficheros.

El camino o path de un fichero o directorio es la secuencia de directorios que se ha de recorrer para acceder a un determinado fichero separados por /. Siguiendo el ejemplo que se visualiza en la Fig. 4.2 de la pág. 73, existen dos formas del path o camino:

- el camino absoluto que muestra toda la ruta a un fichero, `/home/luis/Carta.txt`.
- el path relativo a un determinado directorio, por ejemplo si no encontramos en el directorio `/home`, el path relativo al fichero `Carta.txt` es `luis/Carta.txt`

Todos los directorios contienen dos directorios especiales:

- El directorio actual, representado por el punto `.`
- El directorio padre representado por dos puntos `..`

Existen 3 ficheros estándar implementados en UNIX y que es importante conocer para realizar determinadas acciones:

- Entrada estándar (stdin): Teclado (0).
- Salida estándar (stdout): Pantalla (1).

- Errores estándar (stderr): Pantalla (2).

La redirección entrada/salida, que se explicará más adelante, permite cambiar estas asignaciones en cualquier momento.

## 4.4 Estructura del Sistema de Ficheros

Existe una jerarquía de directorios que para un sistema estándar sería:

/ Raíz del sistema de archivos.

- **/dev**: fichero especiales de dispositivos conectados al ordenador.
- **/lib**: bibliotecas del sistema, necesarias para que se ejecuten los programas que residen en **/bin**.
- **/bin**: órdenes más empleadas.
- **/etc**: datos y órdenes restringidas al superusuario, reservado para los ficheros de configuración del sistema. En este directorio no debe aparecer ningún programa.
- **/tmp**: ficheros temporales, se borran periódicamente.
- **/proc**: ficheros especiales que, o bien reciben, o bien envían información al kernel del sistema.
- **/sbin**: programas que son únicamente accesibles al superusuario o root.
- **/usr**: órdenes, bibliotecas y programas adicionales.
  - **/usr/lib**: librerías generales de los programas.
  - **/usr/bin**: programas de uso general.
  - **/usr/man**: manuales accesibles con el comando `man`.
  - **/usr/doc**: documentación general del sistema.
  - **/usr/etc**: ficheros de configuración generales.
  - **/usr/include**: ficheros de cabecera de C/C++ (.h).
  - **/usr/info**: ficheros de información de GNU.

- `/usr/sbin`: programas de administración del sistema.
- `/usr/src`: código fuente de programas.
- `/users` Directorios de usuarios (puede aparecer también como `/home`).

Los ficheros se especifican por `{camino jerárquico}/nombre{.ext}`

donde las llaves`{}` pueden ser o no necesarias. Los ficheros pueden constar de una extensión que es lo que aparece tras el punto.

El directorio raíz es el único que no tiene nombre. Cada usuario tiene un directorio **HOME** que es el directorio asignado a ese usuario para que almacene sus ficheros. El camino de este directorio está contenido en la variable **HOME**.

## 4.5 Acceso al Sistema de Ficheros

El procedimiento a emplear para acceder a la información almacenada en los distintos sistemas de almacenamiento de un ordenador no es tan sencilla como en Windows, y requiere un proceso llamado “montado”. Cuando se ha terminado de trabajar con un determinado dispositivo hay que “desmontarlo”.

Por ejemplo el proceso para leer un disquete sería el siguiente:

1. Introducir el disquete en la disquetera.
2. Montar el sistema de archivos del mismo.
3. Leer, grabar, y manipular el contenido del disquete.
4. Desmontar el sistema de archivos del disquete.
5. Extraer el disquete de la disquetera.

El proceso puede parecer complejo pero es el precio a pagar por la seguridad, puesto que de esta forma se garantiza que no exista ninguna aplicación que esté usando el disquete cuando se extraiga. (En el caso de los CD-ROM Linux impide su extracción hasta que se desmonta).

## 4.6 Permisos

Linux, al igual que todos los sistemas Unix, mantiene un sistema de permisos de acceso a los ficheros muy estricto, a fin de controlar qué es lo que se puede hacer con ellos, y quien lo puede hacer. Estos permisos se identifican con letras y son:

- *r*: permiso de lectura del fichero.
- *w*: permiso de escritura en el fichero.
- *x*: permiso de ejecución del fichero.
- *s*: permiso para cambiar el propietario del fichero.

Al contrario que en Windows o MS-DOS los programas ejecutables de Linux no están marcados por una determinada extensión (.exe) sino por un atributo, el permiso de ejecución *x*. Si se elimina este atributo a un programa, Linux no será capaz de ejecutarlo.

A su vez cada uno de estos permisos se aplica:

- Al dueño del fichero (*u*).
- Al grupo de usuarios al que pertenece el dueño (*g*).
- Al resto de usuarios (*a*).

Así un fichero determinado puede tener permiso para ser leído, escrito y ejecutado por su dueño, leído y ejecutado por el grupo al que pertenece y no tener ningún tipo de acceso para los demás usuarios. Como se puede entender este tipo de mecanismo es especialmente útil cuando se trabaja en grupo en un determinado proyecto.

## 4.7 Comandos de Manipulación

A continuación se explican las funciones de los comandos más utilizados que permiten manipular el contenido de los ficheros.

### 4.7.1 Listado del Contenido de Directorios: `ls`

Una de las acciones más habituales a la hora de trabajar es mostrar el contenido de un directorio, como se verá en el Capítulo referente a los Componentes de Linux, existen herramientas gráficas con este fin, no obstante el shell incluye un programa con este mismo fin:

- `ls`: muestra los nombres de los ficheros y subdirectorios contenidos en el directorio en el que se está. Sólo se obtienen los nombres de los ficheros, sin ninguna otra información.
- `ls -a`: muestra todos los ficheros incluyendo algunos que ordinariamente están ocultos para el usuario (aquellos que comienzan por un punto). Se recuerda que el fichero punto `.` indica el directorio actual y el doble punto `..` el directorio padre, que contiene, al actual.
- `ls -l`: es la opción de lista larga; muestra toda la información de cada fichero incluyendo: protecciones, tamaño y fecha de creación, o del último cambio introducido, entre otros.
- `ls -c`: muestra ordenando por día y hora de creación.
- `ls -t`: muestra ordenando por día y hora de modificación.
- `ls -r`: muestra el directorio y lo ordena en orden inverso.
- `ls subdir`: muestra el contenido del subdirectorio `subdir`.
- `ls -l filename`: muestra toda la información sobre el fichero.
- `ls --color`: muestra el contenido del directorio coloreado.

Las opciones anteriores pueden combinarse. Por ejemplo:

- `ls -cr`: muestra el directorio ordenando inversamente por fechas.

El comando `ls` admite los caracteres de sustitución o metacaracteres (`*`) y (`?`). El carácter `*` representa cualquier conjunto o secuencia de caracteres. El carácter `?` representa cualquier carácter, pero sólo uno.

#### 4.7.2 Creación de Subdirectorios: **mkdir**

El comando **mkdir** (make directory) permite a cada usuario crear un nuevo subdirectorio:

```
mkdir subdir1
```

donde **subdir** es el nombre del directorio que se va a crear.

#### 4.7.3 Borrado de Subdirectorios: **rmdir**

Este comando borra uno o más directorios del sistema (remove directory), siempre que estos subdirectorios estén vacíos. Por ejemplo:

```
rmdir subdir1
```

donde **subdir** es el nombre del directorio que se va a eliminar.

#### 4.7.4 Cambio de Directorio: **cd**

Este comando permite cambiar de directorio a partir del directorio actual de trabajo:

- **cd dire:** traslada al subdirectorio **dire** (que deberá existir como subdirectorio en el directorio actual).
- **cd ..:** retrocede un nivel en la jerarquía de directorios. Al contrario que en MS-DOS en Linux no existe la forma **cd..** (sin espacio entre **cd** y los dos puntos).
- **cd:** sitúa nuevamente en el directorio personal del usuario.

#### 4.7.5 Situación Actual: **pwd**

El comando **pwd** visualiza o imprime la ruta del directorio en el que se encuentra en un momento determinado. Este comando es uno de los pocos que no tiene opciones y se utiliza escribiendo simplemente **pwd**.



#### 4.7.6 Acceso a Unidades de Disco: mount-umount

Linux a diferencia de Windows no utiliza letras (a:, c:, d:... ) para acceder a las distintas unidades de disco de un ordenador. En Linux para acceder al contenido de una unidad de disco o de un CD-ROM este tiene que haber sido previamente “montado”. El montado se realiza mediante el comando *mount*, con lo que el contenido de la unidad se pone a disposición del usuario en el directorio de Linux que se elija. Por ejemplo para acceder al CD-ROM se teclearía el siguiente comando:

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

`-t iso9660` indica el tipo de sistema que usa la unidad de disco para guardar los ficheros (las más usuales son: `iso9660` en el caso de un CD-ROM, `vfat` en el caso de Windows, y `ext2` en el caso de Linux); `/dev/cdrom` indica el dispositivo que se va a montar.

`/mnt/cdrom` es el directorio en el que se pondrá a disposición del usuario el contenido del CD-ROM. Para montar disquetes se suele utilizar el directorio `/mnt/floppy`.

Cuando el usuario haya dejado de usar ese disco deberá “desmontarlo” mediante el comando `umount` antes de sacar el disquete o el CD-ROM. En este caso debería escribir:

```
umount /mnt/cdrom
```

Para utilizar el comando `mount` de la forma anterior hace falta ser administrador o root. Para que un usuario común pueda utilizar disquetes, o CD-ROM, hay que editar el fichero `/etc/fstab`.

Una vez seguidos los pasos anteriores cualquier usuario podrá “montar” un disquete escribiendo el comando `mount /mnt/floppy`.

Existen en la actualidad distribuciones (p. ej. Linux Mandrake) que realizan este proceso de forma automática por lo que las unidades de disquete y CD-ROM quedan accesibles a todos los usuarios de una forma sencilla, empleando los comandos:

```
mount /mnt/floppy
```

```
umount /mnt/floppy
```

 siempre que `/mnt/floppy` sea la ruta adecuada.

#### 4.7.7 Copia de Ficheros: `cp`

Este comando tiene la forma `cp file1 file2`.

Crea una copia de `file1` y le llama `file2`. Si `file2` no existe, lo crea con los mismos atributos de `file1`. Si `file2` ya existe, su contenido queda destruido y es sustituido por el de `file1`. El fichero `file2` estará en el mismo directorio que `file1`. Tanto `file1` como `file2` indican el nombre de un archivo, que puede incluir la ruta al mismo si alguno de ellos no se encuentra en el directorio actual.

Otra posibilidad es `cp file1 file2 namedir` que realiza copias de `file1` y `file2` en el directorio `namedir`.

#### 4.7.8 Traslado y Cambio de Nombre de Ficheros: `mv`

Este comando tiene una forma similar al anterior: `mv file1 file2`

Realiza la misma función que el anterior pero además destruye el fichero original. En definitiva traslada el contenido de `file1` a `file2`; a efectos del usuario lo que hace es cambiar el nombre a `file1`, llamándole `file2`.

De igual forma, `mv file1 file2 namedir` traslada uno o más ficheros al directorio `namedir` conservándoles el nombre.

El comando `mv namedir1 namedir2` cambia el nombre del subdirectorio `namedir1` por `namedir2`.

Cabe aclarar que el comando `mv` sirve así mismo para cambiar el nombre de los ficheros.

#### 4.7.9 Enlaces a Ficheros: `ln`

En Linux un mismo fichero puede estar repetido con más de un nombre, ya que con el comando `cp` se pueden realizar cuantas copias se desee del fichero. Pero, a veces, es más práctico tener un mismo fichero con varios nombres distintos, y lo que es más importante, poder acceder a él desde más de un directorio. En Linux esto recibe el nombre de *enlaces múltiples a un fichero*. El ahorro de espacio de disco es importante al poder compartir un fichero más de un usuario. Estos enlaces son muy prácticos a la hora de utilizar ficheros

que pertenecen a directorios distintos. Gracias a los enlaces se puede acceder a muchos ficheros desde un mismo directorio, sin necesidad de copiar en ese directorio todos esos ficheros. La forma de este comando es:

```
ln file1 file2
```

A partir de este momento el fichero `file1` tiene dos nombres: `file1` y `file2`. A diferencia de los comandos `cp` y `mv`, este comando toma más precauciones, ya que advierte previamente si el nombre `file2` está ocupado, y en este caso no se ejecuta.

Los ficheros enlazados a otro se borran como los ficheros normales. Si se borra el fichero original permanece su contenido en los ficheros enganchados.

#### 4.7.10 Borrado de Ficheros: `rm`

Este comando tiene la siguiente forma:

```
rm file1 file2
```

Elimina uno o más ficheros de un directorio en el cual se tenga permiso de escritura. Con este comando resulta facilísimo borrar ficheros inútiles, y desgraciadamente, también los útiles. Por eso es conveniente y casi imprescindible emplear la opción `-i`, de la forma siguiente:

```
rm -i file1 file2
```

Con esta opción, Linux pedirá confirmación para borrar cada fichero de la lista, de si realmente se desea su destrucción o no. Se recomienda usar siempre este comando con esta opción para evitar el borrado de ficheros útiles.

#### 4.7.11 Características de un Fichero: `file`

Este comando realiza una serie de comprobaciones en un fichero para tratar de clasificarlo. Su formato es:

```
file fich
```

Tras su ejecución este comando muestra el tipo del fichero e información al respecto del mismo.

#### 4.7.12 Cambio de Modo de Ficheros: **chmod**-**chown**-**chgrp**

Los permisos de cada fichero se pueden ver con el comando *ls -l*.

Para cambiar los permisos de un fichero se emplea el comando **chmod**, que tiene el formato siguiente:

```
chmod [quien] oper permiso files
```

**quien** indica a quien afecta el permiso que se desea cambiar. Es una combinación cualquiera de las letras *u* para el usuario, *g* para el grupo del usuario, *o* para los otros usuarios, y *a* para todos los anteriores. Si no se da el quien, el sistema supone *a*.

**oper** indica la operación que se desea hacer con el permiso. Para dar un permiso se pondrá un *+*, y para quitarlo se pondrá un *-*.

**permiso** indica el permiso que se quiere dar o quitar. Será una combinación cualquiera de las letras anteriores : *r,w,x,s*.

**files** indica los nombres de los ficheros cuyos modos de acceso se quieren cambiar.

Los permisos de lectura, escritura y ejecución tienen un significado diferente cuando se aplican a directorios y no a ficheros normales. En el caso de los directorios el permiso *r* significa la posibilidad de ver el contenido del directorio con el comando *ls*; el permiso *w* da la posibilidad de crear y borrar ficheros en ese directorio, y el permiso *x* autoriza a buscar y utilizar un fichero concreto.

Por otra parte, el comando **chown** se emplea para cambiar de propietario a un determinado conjunto de ficheros. Este comando sólo lo puede emplear el actual propietario de los mismos. Los nombres de propietario que admite Linux son los nombres de usuario, que están almacenados en el fichero */etc/passwd*. La forma general de este comando es la siguiente:

```
chown newowner file1 file2 ...
```

Análogamente, el grupo al que pertenece un fichero puede ser cambiado con el comando **chgrp**, que tiene una forma general similar a la de **chown**:

```
chgrp newgroup file1 file2...
```

#### 4.7.13 Espacio Ocupado en el Disco: **du-df**

El comando **du** permite conocer el espacio ocupado en el disco por un determinado directorio y todos los subdirectorios que cuelgan de él. Para usarlo basta simplemente colocarse en el directorio adecuado y teclear **du**, con lo cual se obtiene el espacio de disco utilizado en bloques. Para obtener la información en bytes se debe emplear el comando con la opción **h:du-h**.

El comando **df** por el contrario informa del espacio usado por las particiones del sistema que se encuentren montadas.

#### 4.7.14 Visualización de Ficheros sin Formato: **cat**

Este comando permite visualizar el contenido de uno o más ficheros de forma no formateada. También permite copiar uno o más ficheros como apéndice de otro ya existente. Algunas formas de utilizar este comando son las siguientes:

- **cat filename**: saca por pantalla el contenido del fichero **filename**.
- **cat file1 file2...**: saca por pantalla, secuencialmente y según el orden especificado, el contenido de los ficheros indicados.
- **cat file1 file2 >file3**: el contenido de los ficheros **file1** y **file2** es almacenado en **file3**.
- **cat file1 file2 >>file3**: el contenido de **file1** y **file2** es añadido al final de **file3**.
- **cat >file1**: acepta lo que se introduce por el teclado y lo almacena en **file1** (se crea **file1**).

#### 4.7.15 Visualización de Ficheros con Formato: **pr**

Este comando, a diferencia de **cat**, imprime por consola el contenido de los ficheros de una manera formateada, por columnas, controlando el tamaño de página y poniendo cabeceras al comienzo de las mismas. Está muy relacionado con el comando **lp** de salida por impresora. Las formas más importantes que admite son:

`pr file`: produce una salida estándar de 66 líneas por página, con un encabezamiento de 5 líneas (2 en blanco, una de identificación y otras 2 líneas en blanco).

- `pr -ln file`: produce una salida de `n` líneas por página (cuando el tamaño de papel de impresora, por ejemplo, tiene un número de líneas distinto de 66).
- `pr -p file`: hace una pausa para presentar la página, hasta que se pulsa `<return>` para continuar.
- `pr -t file`: suprime las 5 líneas del encabezamiento y las del final de página.
- `pr -wn file`: ajusta la anchura de la línea a `n` posiciones.
- `pr -d file`: lista el fichero con espaciado doble.
- `pr +n file`: imprime el fichero a partir de la página `n`.

Además de los ejemplos anteriores, se pueden combinar varias opciones en un mismo comando.

#### 4.7.16 Visualización de Ficheros Pantalla a Pantalla: `more-less`

Estos comandos permiten visualizar un fichero pantalla a pantalla. El número de líneas por pantalla es de 23 líneas de texto y una última línea de mensajes, donde aparecerá la palabra `more`. Cuando se pulsa la barra espaciadora (el espacio en blanco), se visualizará la siguiente pantalla. Para salir de este comando (terminar la visualización) se pulsa `<ctrl>d` o `q`. Por ejemplo: `more file`

El comando `less` es muy similar al anterior pero permite el desplazamiento a lo largo del texto empleando las teclas de cursores pudiendo desplazarse hacia arriba o abajo de un fichero.

#### 4.7.17 Búsqueda en Ficheros: **grep**

El comando **grep** localiza una palabra, clave o frase en un conjunto de directorios, indicando en cuáles de ellos la ha encontrado. Este comando rastrea fichero por fichero, por turno, imprimiendo aquellas líneas que contienen el conjunto de caracteres buscado. Si el conjunto de caracteres a buscar está compuesto por dos o más palabras separadas por un espacio, se colocará el conjunto de caracteres entre apóstrofes ('). Su formato es el siguiente:

```
grep 'conjuntocaracteres' file1 file2 file3
```

Este comando permite seleccionar, entre todas las líneas de uno o más ficheros, aquellas que contienen un motivo que satisface una expresión regular determinada:

```
grep [-opcion] expresión_regular [referencia...]
```

Las opciones principales son:

- **c**: lo único que se hace es escribir el número de las líneas que satisfacen la condición.
- **i**: no se distinguen mayúsculas y minúsculas.
- **l**: se escriben los nombres de los ficheros que contienen líneas buscadas.
- **n**: cada línea es precedida por su número en el fichero.
- **s**: no se vuelcan los mensajes que indican que un fichero no se puede abrir.
- **v**: se muestran sólo las líneas que no satisfacen el criterio de selección.

#### 4.7.18 Difusión de Programas y Ficheros: **tar-gzip**

Ambos son ampliamente empleados para la difusión de programas y ficheros en Linux. El primero de ellos agrupa varios ficheros en uno solo o archivo, mientras que el segundo los comprime. En conjunto estos dos programas actúan de forma muy similar a programas como Winzip. Para crear un nuevo archivo se emplea:

```
tar cvf nombre_archivo.tar fichero1 fichero2
```

donde `fichero1`, `fichero2` son los ficheros que se van a añadir al archivo `tar`. Si se desea extraer los ficheros se emplea:

```
tar xpvf nombre_archivo.tar fichero1
```

Al contrario que `tar`, `gzip` comprime un único fichero con lo que la información se mantiene pero se reduce el tamaño del mismo. Su uso es muy sencillo:

```
gzip fichero
```

Con esto se comprime fichero (es borrado) y se crea un fichero con nombre `fichero.gz`. Si lo que se desea es descomprimir un fichero y recuperar el fichero inicial, se emplea:

```
gzip d fichero.gz
```

#### 4.7.19 Comando de Impresión: `lpr`

El comando `lpr` se emplea para imprimir una serie de ficheros. Si se emplea sin argumentos imprime el texto que se introduzca a continuación en la impresora por defecto. Por el contrario, `lpr nombre_fichero` imprime en la impresora por defecto el fichero indicado.



## Capítulo 5

# Componentes y Distribuciones Comerciales



### 5.1 Componentes

Para generar un producto útil, las distribuciones de Linux se crean relacionando varios componentes necesarios, como ser el kernel, sistema de archivos, manejadores, servidores (web, de correos, de DNS, etc.), aplicaciones, entre otras.

Los subsistemas son desarrollados típicamente dentro del esquema del OSS, así como varios de ellos tienen una base de código fuente de tamaño/complejidad que excede el Kernel de Linux.

Los componentes externos vienen de muchas fuentes, y son escogidos individualmente por el vendedor de la distribución de un producto en particular. Un fuente frecuente de controversia surge debido a que los vendedores de distribución conectan código que no es GPL con el kernel de Linux y lo distribuyen de manera masiva.

La Tabla 5.1 de la pág. 89 nombra los principales componentes de Linux.

### 5.1.1 Kernel GPL

El Kernel es la parte central de Linux que está manejada expresamente por Linus Torvalds y está protegida vía GPL.

Las funciones contenidas dentro del Kernel de Linux son:

- Características centrales del SO (distribución del tiempo, administración de la memoria, hilos, abstracción del hardware, etc.).
- Stack de red.
- Sistema de archivos.

El Kernel de Linux esta en constante actualización y es el núcleo central de todas las distribuciones Linux.

Las versiones se dividen en 3 categorías: estable, alpha y beta.

Las versiones estables, son las que han sido totalmente probadas y son la base de todas las distribuciones.

Las versiones alpha son las que están en una etapa previa de evaluación entre la beta y la estable y son las candidatas a la próxima versión estable.

Las versiones beta son versiones estrictamente de evaluación, en la que se incorporan mejoras o novedades que deben ser debidamente evaluadas, antes de pasar a la etapas de alpha y estable.

Componente	Código Fuente Base	Encargado(s) de Proporcionarlo
Kernel	SO Básico, Conexión en Red, Stacks	Linux ( <a href="http://www.kernel.org/">http://www.kernel.org/</a> )
Sistema de Archivos	MSDOS, EXT2FS	Kernel de Linux
Biblioteca de System	Glibc, Lib5c	GNU/FSF
Manejadores		Contribuidores Inidiv., Linux
Herramientas para Usuarios	Herramientas para Usuarios de GNU	GNU/FSF
Instalación del System	LISA	Caldera
Instalación de Aplicaciones Mantenimiento	Red Hat Package Manager	Red Hat
Herramientas de Desarrollo	Herramientas de Desarrollo GNU GCC	GNU/FSF
Servidor de Web	APACHE	El Grupo Apache <a href="http://www.apache.org/">http://www.apache.org/</a>
Servidor de Correos	SendMail	<a href="http://www.sendmail.org/">http://www.sendmail.org/</a>
Servidor de DNS	BIND	<a href="http://www.bind.org/">http://www.bind.org/</a>
Servidor X	Xfree86 / Metro X	Proyecto Xfree86/ Metro X comercial
Manejador de Ventanas	FVWM	GPL
Widgets	Motif	X Consortium
Herramientas de Escritorio	X Contrib, KDE Gnome	X Consortium <a href="http://www.kde.org/">http://www.kde.org/</a> <a href="http://www.gnome.org/">http://www.gnome.org/</a>
Mantenimiento	Paquetes RPM Instalados Específico de cada distribución	Red Hat (gratuito)  Debian / Slackware

Tabla 5.1: Componentes de Linux.

### 5.1.2 Manejadores GPL

Formado por un conjunto variado de módulos para funciones y dispositivos estándar. Además, frecuentemente se incluye una selección de módulos no estándar.

Los manejadores de dispositivos de Linux son típicamente desarrollados por usuarios de dispositivos específicos sobre sus máquinas. Este proceso acumulativo de poquito en poquito ha creado un gran conjunto de manejadores de dispositivos de:

- Vídeo.
- Red.
- PCMCIA.

### 5.1.3 Librerías de Sistema y Aplicaciones GNU GPL

Las librerías de sistema proporcionan:

- API POSIX básicos para servicios de sistema.
- API básicos para utilerías de líneas de comando/shell.

Las librerías de sistemas dentro de una distribución de Linux **no** son manejadas por Linux. Como tales, se han presentado en esta área dos librerías dominantes *glibc* y *lib5c* quienes introducen incompatibilidades menores entre diferentes aplicaciones.

### 5.1.4 Herramientas del Usuario (GPL, GNU FSF)

Estas son herramientas de línea de comando de UNIX y ambientes de shell básicos. Existen muchos ambientes de shell, aunque todos tienen soporte para FSF.

También, dentro de esta categoría están las viejas aplicaciones “stand by” tales como Finger, Telnet, etc.

### 5.1.5 Herramientas de Desarrollo (GPL)

Uno de los éxitos del sistema operativo UNIX es la disponibilidad gratuita de herramientas de desarrollo/compilación. Los compiladores de lenguaje GCC y PERL son frecuentemente proporcionados de manera gratuita para todas las versiones de Linux y están disponibles también para otras variantes de UNIX.

### 5.1.6 X Server

El X Server pertenece a MIT, quien se lo concede a la X Consortium. Las prácticas de licenciamiento de X Consortium son vistas como demasiado limitantes, por lo que se lanzaron una serie de iniciativas públicas X. Entre las que se lanzaron, el Xfree86 se convirtió en la distribución dominante.

La configuración del sistema Xfree86 sobre Linux puede ser un proceso bastante difícil, que consume mucho tiempo. Linux no tiene capa de abstracción de hardware para los servicios de video, y la mayor parte de los fabricantes de tarjeta de video no proporcionan manejadores de video para el SO Linux. Así, Xfree86 proporciona soporte interno para una amplia variedad de tarjetas de video y chipsets. Una configuración correcta de Xfree86 requiere de que el usuario conozca al fabricante, modelo y chipset de su tarjeta de video. En muchos casos, el usuario deberá conocer o calcular la sincronización de video.

### 5.1.7 Widgets y Desktops

Existen muchos conjuntos de widget (interfaces) que existen en muchas aplicaciones, por lo que no todas las aplicaciones X parecen ser la misma o actúan de la misma manera que Windows. Se le considera a **Motif** como el conjunto de widget de UNIX de facto, pero debido a que no se puede distribuir libremente, ésta es contraria al modelo de Linux.

Por lo tanto, las distribuciones de Linux generalmente escogen uno de tantos conjuntos Widget, aunque no sean completamente compatibles:

- Motif.
- LessTif.
- Xaw3d.

- Qt.

Obviamente, este caos ha entorpecido varios esfuerzos para unificar el escritorio, así como los conjuntos de widget. A la típica manera Linux, existen varios esfuerzos que compiten entre sí:

- Gnome.
- KDE.
- FreeQT / KDE.
- CDE / Comercial.

#### 5.1.8 Servidor WEB Apache



Apache es un “programa de servidores”, que son el cerebro y corazón de poderosos ordenadores que administran los sitios Web.

Contrariamente a los programas comerciales, distribuidos en determinados códigos ejecutable, Apache es un programa libre (freeware). Está publicado sobre Internet en un código original, el código fuente, que cada uno es libre de rescribir a su gusto para adaptar mejor el programa a sus necesidades específicas o rectificar errores.

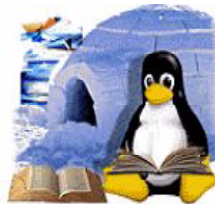
La única obligación es que esta versión de Apache debe ser publicada a su turno sobre Internet en código fuente.

Forma así una comunidad de un potencial de millones de profesionales que participan en el desarrollo del programa. Apache compite directamente con un servidor desarrollado por Netscape y con un producto de Microsoft, Internet Information Server (IIS).

Se basó originalmente en codificación e ideas basadas en el servidor HTTP mas popular de todos, el NCSA httpd 1.3 (principios de 1995). Esto ha desencaminado en un sistema que puede rivalizar y por que no, sobrepasar a casi cualquier otro servidor basado en UNIX HTTP en cuanto a funcionalidad, eficacia y rapidez. Desde su comienzo, se ha vuelto a escribir completamente, e incluye muchos rasgos nuevos.

Apache ha mostrado ser substancialmente mas rápido que muchos otros servidores libres. Aunque seguro que los servidores comerciales han exigido superar la rapidez del Apache, ellos opinan que es mejor tener un servidor libre y en su mayor parte rápido, que un servidor extremadamente rápido pero que cueste miles de dólares. Apache funciona en sitios que tienen millones de usos al día, y estas se ejecutan sin complicaciones.

#### 5.1.9 Servidor de Correo Sendmail



Sendmail es el agente de transporte de correo más común de Internet (en los sistemas UNIX). Aunque actúa principalmente como MTA, también puede ser utilizado como MUA (aunque no posee interfaz de usuario). Las misiones básicas de sendmail son las siguientes:

- Recolección de mails provenientes de un Mail User Agent (MUA) como pueden ser elm, Eudora o pine; o provenientes de un Mail.
- Transport Agent (MTA) como puede ser el propio sendmail.
- Elección de la estrategia de reparto de los mails, basándose en la información de la dirección del destinatario contenida en la cabecera:
  - Si el mail es local, enviará el mail al programa de reparto local de mails.

- Si el mail no es local, sendmail utilizará el DNS del sistema para determinar el host al que debe ser enviado el mail. Para transferir el mensaje, iniciará una sesión SMTP con el MTA de dicho host.
- Si por alguna razón no es posible mandar el mail a su destino, sendmail almacenará los mails en una cola de correo, y volverá a intentar el envío del mail un tiempo después. Si el mail no puede ser enviado tras un tiempo razonable, el mail será devuelto a su autor con un mensaje de error. Sendmail debe garantizar que cada mensaje llegue correctamente a su destino, o si hay error este debe ser notificado (ningún mail debe perderse completamente).
- Utilización de “alias” entre los usuarios del sistema; lo que permitirá (entre otras funciones) crear y mantener listas de correo entre grupos.
- Ejecución como agente de usuario (MUA). Aunque no posee interfaz de usuario, sendmail también permite el envío directo de mails a través de su ejecutable.

Todas estas características y muchas otras que posee el sendmail deben ser configuradas y variarán de unos sistemas a otros. Para configurarlas se utiliza el fichero de configuración de sendmail. La revisión y modificación de este fichero es bastante complicada y se necesita de una serie de conocimientos previos.

#### 5.1.10 Servidor DNS Bind

En una red TCP/IP típica, el servidor de nombres es una máquina encargada de hacer la conversión entre direcciones nombres de máquinas y direcciones IP y viceversa. El software utilizado para este fin fue desarrollado en la universidad de Berkeley, California, y es llamado por esta razón BIND (Berkeley Internet Name Domain).

El software BIND incluye un demonio llamado `named`, así como utilerías que permiten controlar este programa, que es el encargado de realizar las traducciones de nombres a IP.

Este software puede compilarse en casi cualquier versión moderna de Unix, aunque es necesario leer cuidadosamente las instrucciones que acompañan al código fuente para hacer las adecuaciones necesarias a cada ambiente.



Para algunas versiones del sistema operativo Linux es posible conseguir paquetes precompilados que facilitan en gran manera la instalación del software. El proceso de instalación varía de una versión de Unix a otra.

El sistema operativo debería arrancar por sí solo el demonio `named` al terminar su propia secuencia de arranque. En los sistemas UNIX BSD (Linux Slakware, por ejemplo), esto se logra incluyendo en el archivo `rc.local` la línea:

```
named &
```

Los sistemas que, por el contrario, siguen la norma System V para los archivos de arranque, contienen directorios especiales con scripts de shell que levantan los programas necesarios en cada runlevel. El caso de Linux RedHat, es el más sencillo: la misma instalación mediante el comando `rpm` agrega en el directorio correspondiente el script necesario para levantar el demonio.

## 5.2 Distribuciones Comerciales



Una **Distribución** es un agrupamiento del núcleo del sistema operativo Linux y otra serie de aplicaciones de uso general o no tan general. En principio las empresas que desarrollan las distribuciones de Linux están en su derecho al cobrar una cierta cantidad por el software que ofrecen, aunque en la mayor parte de las ocasiones se pueden conseguir estas distribuciones desde Internet, de revistas o de amigos, siendo todas estas formas gratuitas y legales.

Las distribuciones más conocidas son *RedHat*, *Debian*, *Slackware*, *SuSE* y *Corel Linux*, todas ellas incluyen el software más reciente y empleado lo cual incluye compiladores de C/C++, editores de texto, juegos, programas para el acceso a Internet, así como el entorno gráfico de Linux: X Window.

### Componentes Fundamentales de una Distribución

Aunque la esencia de Linux es el kernel, se necesita mucho mas para convertirlo en un ambiente operativo completo. Los elementos clave de una típica distribución Linux son:

- **Boot Manager:** el boot manager o administrador de inicio es una utilidad que reside en el sector de inicio maestro, que el bios de la computadora carga al inicio. Si se agrega Linux a un sistema operativo con Windows u otro SO, el administrador de inicio permite seleccionar el SO que se quiere ejecutar. Aunque existen administradores de inicio de otros fabricantes, como el System commander de V Communications, Linux incluye su propio administrador de inicio, conocido como LILO (Linux Loader), que por lo general se instala por omisión.
- **Interfaz de Usuario:** en el núcleo, Linux es un derivado de Unix, y su interfaz por omisión es una línea de comando DOS. Para hacer más accesible a Linux, hay varios ambientes de escritorio gráficos que pueden semejarlo al SO Windows o Macintos. Algunos ejemplos son CDE (Common Desktop Environment), KDE (Kool Desktop Environment) y GNOME (GNU Network Object Model Environment).
- **Sistema X Window:** al igual que GDI (Graphics Device Interface) de windows, X window es el subsistema gráfico que soporta una interfaz de usuario gráfica. Uno de los servidores más comunes es Xfree86. Durante la configuración se puede instalar o no un sistema X Window, según se prefiera.
- **Servicio de Internet:** Linux posee soporte nativo para TCP/IP y para todos los protocolos internet comunes, como DNS (Domain Name System), http (web server), SMTP (Correo Electrónico), FTP (File Transport Protocol), TNP (Foros de Discusión) y PPP/SLIP (para Cuentas Conmutadas). Gran parte de las distribuciones incluyen Apache, el popular servidor web de código público, así como visualizadores web, clientes de correo electrónico, lectores de foros y otras aplicaciones de internet.
- **Servicios de Impresión de Archivos:** estos servicios permiten a un sistema acceder a los recurso de red y compartir archivos e impresoras con otros usuarios. Entre los servicios típicos se incluyen NFS (Netwrk File System) para compartir archivos con otros sistemas basados en Unix; y

Samba, que hace que los sistemas Linux actúen como servidores Windows NT y Appletalk.

- Aplicaciones: Linux aún no goza del mismo soporte de aplicaciones que Windows. Pero gran parte de las distribuciones incluyen, literalmente cientos de aplicaciones, y si deseas buscarlas, encontrará una buena variedad de herramientas en otro lado. Además de editores básicos de texto como Crisp y Xemacs, hay diversas aplicaciones y suites de productividad. GIMP (GNU Image Manipulation Program) es un clon de Adobe Photoshop.
- Administradores de Paquetes: los administradores de paquete son herramientas diseñadas para instalar, desinstalar y actualizar aplicaciones en sistemas Linux. El Debian Package Management System y el Red Hat Package Manager son administradores de paquetes bien conocidos.
- Herramientas del Programador: puesto que Linux es un favorito de los programadores, no sorprende que las distribuciones incluyan muchas herramientas de programación. Entre estas se incluyen utilerías para administrar y crear programas a partir de código fuente, eliminadores de fallas como gdb, lenguajes de programación y lenguajes de guiones como Pitón y TCL.
- Bibliotecas: Linux incluye un conjunto central de rutinas de bibliotecas de vínculo dinámico que se encuentran en Windows, las libcs son módulos ejecutables que se encuentran en Windows y realizan funciones ocultas que soportan a otras aplicaciones. Gran parte de las distribuciones incluyen muchas bibliotecas; los tipos más comunes son GNU *Libc(glibc)* y *Libc5*.



# Bibliografía

- [1] J.García Galón; I. Aguinaga; A. Mora. *Aprenda Linux como si estuviera en primero*. Universidad de Navarra, San Sebastián-España, 2000.
- [2] IBM Press. *Linux Executive Report*. IBM Press, USA, 2004.
- [3] W. Stallings. *Unix*. Prentice Hall, NJ-USA, 2001.
- [4] M. Welsh. *Linux: Installation and Getting Started*. USA, 1996.



# Índice de Materias

- Apache, 92
- Aplicación
  - archiver, 54
  - audio mixer, 59
  - calendar, 66
  - cd player, 59
  - change password, 66
  - color xterm, 59
  - diskfree, 66
  - file manager, 57
  - gedit, 67
  - gnomecard, 66
  - gnnumeric spreadsheet, 67
  - gq view, 66
  - icon editor, 54
  - kdehelp, 48
  - kedit, 47
  - kfind, 48
  - kfm, 41
  - konsole, 46
  - korganizer, 54
  - kview, 54
  - kwrite, 47
  - paint, 54
  - ps viewer, 54
  - regular xterm, 59
  - search tool, 58
  - simple calculator, 66
  - snapshot, 54
  - system info, 66
  - system monitor, 66
  - time tracking tool, 67
- Arranque, 27
- Caracteres
  - especiales, 15
  - válidos, 14
- Comando
  - cat, 83
  - cd, 78
  - chgrp, 82
  - chmod, 82
  - chown, 82
  - cp, 80
  - df, 83
  - du, 83
  - file, 81
  - grep, 85
  - gzip, 85
  - less, 84
  - ln, 80
  - lpr, 86
  - ls, 77
  - mkdir, 78
  - mke2fs, 31
  - mkswap, 30
  - more, 84
  - mount, 79
  - mv, 80
  - pr, 83
  - pwd, 78
  - rm, 81
  - rmdir, 78

- tar, 85
- umount, 79
- Componentes
  - aplicaciones, 90
  - desktops, 91
  - herramientas
    - de desarrollo, 91
    - de usuario, 90
  - kernel, 88
  - librerías de sistema, 90
  - manejadores, 90
  - servidor de correo, 93
  - servidor DNS, 94
  - servidor web, 92
  - widgets, 91
  - x server, 91
- Distribución Comercial, 10
  - componentes, 96
    - administrador de paquetes, 97
    - aplicaciones, 97
    - bibliotecas, 97
    - boot manager, 96
    - herramientas del programador, 97
    - interfaz de usuario, 96
    - servicio de impresión, 96
    - servicio de internet, 96
    - x window, 96
  - Corel Linux, 10
  - Debian, 10
  - RedHat, 10
  - Slackware, 10
  - SuSe, 10
- DLL, 5
- Enlace
  - Application, 45
  - file system device, 44
  - ftp url, 45
  - mime type, 45
  - tipos de, 44
- Estructura
  - kernel, 13
  - núcleo, 13
  - shell, 13
- Ficheros
  - comandos de manipulación, 76
  - directorio, 73
  - enlaces, 71
  - extensión, 70
  - nombres, 70
  - path, 71
  - sistema de, 16
  - tipos, 16, 70
- GNOME, 55
  - aplicaciones, 57
  - centro de control, 59
  - configuración, 59
  - editor de menús, 65
  - panel, 56
  - pantalla, 56
- Grupos, 12
- Instalación, 27
  - fdisk, 26
  - fips, 22
  - LILO, 22
  - particiones, 24
  - pasos básicos, 23
  - problemas, 32
  - swap, 23
- KDE, 39
  - antecedentes, 40
  - características, 39
  - centro de control, 53
  - configuración, 50
  - editor de menús, 50



- enlaces, 44
  - escritorio virtual, 41
  - pantalla, 40
- Kernel
  - concepto, 13
  - tareas, 13
- Licencia
  - GPL, 8
  - Shareware, 9
- LILO, 22, 31
  - instalación, 31
- Linux
  - antecedentes, 2
  - características, 3
  - compatibilidad, 10
  - componentes, 87
  - conectividad, 10
  - distribuciones comerciales, 87, 95
  - entorno de trabajo, 11
  - entornos gráficos, 39
  - estructura general, 13
  - instalación, 21
  - licencia, 8
  - recursos, 9
  - relación con Unix, 2
  - sistema de ficheros, 69
  - ventajas, 7
- Manejadores
  - de dispositivos, 28
- Memoria Virtual, 4
- Metacaracteres, 15
- Multitarea, 4
- Multiusuario, 4
- Particiones
  - clases, 24
  - crear, 29
- Pipes, 18
- Planificación Mixta, 4
- Problemas
  - con el arranque, 33
  - con el hardware, 34
  - con el software, 37
- Redireccionamiento
  - bifurcación o T, 19
  - de E/S, 17
  - encadenado, 18
  - simple, 17
- Reparticionamiento, 25
- Samba, 11
- Sendmail, 93
- Servidor
  - de correo, 93
  - DNS Bind, 94
  - web, 92
- Shell, 11
  - concepto, 14
  - tipos de, 14
- Sistema
  - entrada y salida, 19
- Sistema de Archivos
  - según S.O., 22
- Sistema de Ficheros
  - acceso, 75
  - crear, 30
  - estructura, 74
  - introducción, 69
  - permisos, 76
- Software
  - gratuito, 1
  - instalación, 31
  - libre, 8
- Swap, 23, 25, 30
- Tuberías, 18
- Unix, 7

- Usuarios, 12
  - categorías de, 12
  - permisos asociados, 12
- X Window, 11