



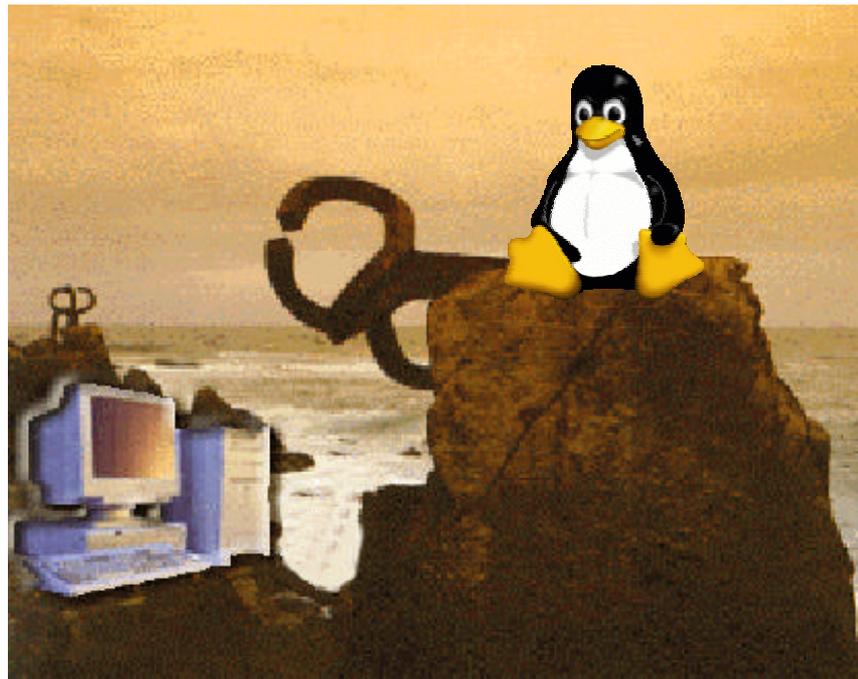
Escuela Superior de Ingenieros Industriales  
Industri Injineruen Goimailako Eskola

UNIVERSIDAD DE NAVARRA - NAFARROAKO UNIBERTSITATEA

# Aprenda LINUX

*como si estuviera en primero*

San Sebastián, Enero 2000



Javier García de Jalón • Iker Aguinaga • Alberto Mora



# Aprenda LINUX

*como si estuviera en primero*

Javier García de Jalón

Iker Aguinaga

Alberto Mora

Pertenece a la colección : *"Aprenda ..., como si estuviera en primero"*



**ÍNDICE**

<b>1.</b>	<b>INTRODUCCIÓN.....</b>	<b>1</b>
1.1	¿QUÉ ES <i>LINUX</i> ?.....	1
1.2	¿QUÉ SON LAS DISTRIBUCIONES?.....	2
1.3	ENTORNO DE TRABAJO: EL SHELL Y X <i>WINDOW</i> .....	2
1.4	USUARIOS Y GRUPOS.....	2
<b>2.</b>	<b>INSTALACIÓN.....</b>	<b>3</b>
<b>3.</b>	<b>INICIACIÓN A <i>LINUX</i>.....</b>	<b>5</b>
3.1	ENTRADA Y SALIDA DEL SISTEMA.....	5
<b>4.</b>	<b>ESTRUCTURA DEL SISTEMA DE ARCHIVOS DE <i>LINUX</i>.....</b>	<b>6</b>
4.1	ARCHIVOS: TIPOS.....	6
4.2	ENLACES.....	6
4.3	EL CAMINO O <i>PATH</i> .....	7
4.4	ESTRUCTURA DEL SISTEMA DE ARCHIVOS DE <i>LINUX</i> .....	7
4.5	ACCESO A LOS DIFERENTES SISTEMAS DE ARCHIVOS.....	9
4.6	PERMISOS.....	10
<b>5.</b>	<b>X <i>WINDOW</i>.....</b>	<b>11</b>
<b>6.</b>	<b>GNOME.....</b>	<b>12</b>
6.1	INICIACIÓN A GNOME.....	17
6.2	APLICACIONES AUXILIARES DE GNOME.....	18
6.2.1	<i>File Manager</i> .....	18
6.2.2	<i>GNOME Search Tool</i> .....	20
6.2.3	<i>Color Xterm</i> , <i>GNOME Terminal</i> y <i>Regular Xterm</i> .....	20
6.2.4	<i>Multimedia</i> .....	20
6.2.5	<i>Otras aplicaciones</i> .....	21
6.3	CONFIGURACIÓN DE GNOME.....	24
<b>7.</b>	<b>KDE.....</b>	<b>31</b>
7.1	PARTES DE LA PANTALLA.....	31
7.2	ADMINISTRACIÓN DE ARCHIVOS, KFM.....	33
7.2.1	<i>Navegar por la estructura de directorios y ver el contenido de los ficheros</i> .....	33
7.2.2	<i>Crear un nuevo directorio</i> .....	35
7.2.3	<i>Borrar un documento o directorio</i> .....	35
7.2.4	<i>Copiar y mover un documento o directorio</i> .....	35
7.2.5	<i>Enlaces de KDE</i> .....	36
7.2.6	<i>Asociar un nuevo tipo de archivo</i> .....	37
7.2.7	<i>Propiedades de un fichero o directorio</i> .....	37
7.2.8	<i>Configura kfm como navegador de Internet</i> .....	38
7.3	APLICACIONES AUXILIARES DE KDE.....	38
7.3.1	<i>konsole</i> .....	38
7.3.2	<i>kedit</i> .....	39
7.3.3	<i>kwrite</i> .....	39
7.3.4	<i>kdehelp</i> .....	39
7.3.5	<i>Kfind</i> .....	40
7.4	CONFIGURACIÓN DE KDE.....	42
7.4.1	<i>Editor de menús</i> .....	42
7.4.2	<i>KDE Control Center</i> .....	45
7.4.3	<i>Añadir aplicaciones al panel</i> .....	46
7.5	OTRAS APLICACIONES DE KDE.....	46
<b>8.</b>	<b>EL SHELL: COMANDOS BÁSICOS DE <i>LINUX</i>.....</b>	<b>48</b>

8.1	INTRODUCCIÓN.....	48
8.2	ALGUNOS COMANDOS SENCILLOS DE <i>LINUX</i> .....	49
8.3	DIRECTORIO PERSONAL.....	50
8.3.1	Listado del contenido de directorios: comando <i>ls</i> .....	50
8.3.2	Creación de subdirectorios. Comando <i>mkdir</i> .....	51
8.3.3	Borrado de subdirectorios. Comando <i>rmdir</i> .....	51
8.3.4	Cambio de directorio. Comando <i>cd</i> .....	51
8.3.5	Situación actual. Comando <i>pwd</i> .....	52
8.3.6	Acceso a unidades de disco.....	52
8.3.7	Copia de ficheros. Comando <i>cp</i> .....	53
8.3.8	Traslado y cambio de nombre de ficheros. Comando <i>mv</i> .....	53
8.3.9	Enlaces a ficheros. Comando <i>ln</i> .....	53
8.3.10	Borrado de ficheros. Comando <i>rm</i> .....	54
8.3.11	Características de un fichero. Comando <i>file</i> .....	55
8.3.12	Cambio de modo de los ficheros comandos <i>chmod</i> , <i>chown</i> y <i>chgrp</i> .....	55
8.4	ESPACIO OCUPADO EN EL DISCO COMANDOS <i>DU</i> Y <i>DF</i> .....	56
8.5	VISUALIZACIÓN SIN FORMATO DE UN FICHERO. COMANDO <i>CAT</i> .....	56
8.6	COMANDO <i>HEAD</i> .....	56
8.7	VISUALIZACIÓN DE FICHEROS CON FORMATO. COMANDO <i>PR</i> .....	56
8.8	VISUALIZACIÓN DE FICHEROS PANTALLA A PANTALLA. COMANDOS <i>MORE</i> Y <i>LESS</i> .....	57
8.9	BÚSQUEDA EN FICHEROS. COMANDOS <i>GREP</i> , <i>FGREP</i> Y <i>EGREP</i> .....	57
8.10	COMANDOS <i>TAR</i> Y <i>GZIP</i> .....	58
8.11	COMANDOS DE IMPRESIÓN.....	59
8.11.1	Comando <i>lpr</i> .....	59
<b>9.</b>	<b>REDIRECCIONES Y TUBERÍAS.....</b>	<b>60</b>
9.1	REDIRECCIONES.....	60
9.2	TUBERÍAS.....	60
9.3	BIFURCACIÓN O T (COMANDO <i>TEE</i> ).....	61
9.4	REDIRECCIÓN DE LA SALIDA DE ERRORES.....	61
<b>10.</b>	<b>EJECUCIÓN DE PROGRAMAS.....</b>	<b>62</b>
10.1	EJECUCIÓN EN EL FONDO & , <i>KILL</i> , <i>NICE</i> Y <i>NOHUP</i> .....	62
10.2	COMANDO <i>TIME</i> .....	62
10.3	COMANDO <i>TOP</i> .....	63
<b>11.</b>	<b>PROGRAMAS DE COMANDOS.....</b>	<b>64</b>
11.1	INTRODUCCIÓN DE COMENTARIOS.....	65
11.2	VARIABLES DEL SHELL.....	65
11.3	COMANDO <i>ECHO</i> .....	65
11.4	PARÁMETROS DE LOS FICHEROS DE COMANDOS.....	66
11.5	OTRAS POSIBILIDADES DE LOS FICHEROS DE COMANDOS.....	67
<b>12.</b>	<b>ANEJOS.....</b>	<b>69</b>
12.1	COMPILADO DE PROGRAMAS EN <i>LINUX</i> .....	69
12.2	COMPILACIÓN Y LINKADO.....	69
12.3	COMANDO <i>MAKE</i> .....	70
12.4	BÚSQUEDA AVANZADA EN FICHEROS. EXPRESIONES REGULARES.....	71
12.4.1	Caracteres especiales.....	71
12.4.2	Expresiones regulares de un solo carácter.....	72
12.4.3	Expresiones regulares generales.....	72
12.5	COMANDOS ÚTILES PARA TRABAJAR EN RED.....	73
12.6	PROTOCOLOS <i>INTERNET</i> ( <i>IP</i> ).....	73
12.7	DENOMINACIÓN SIMBÓLICA DE SISTEMAS <i>INTERNET</i> .....	74
12.8	COMANDO <i>TELNET</i> .....	75
12.9	COMANDO <i>FTP</i> .....	75



## 1. INTRODUCCIÓN

La mayor parte de los ordenadores que existen en la actualidad están diseñados de forma que puedan ejecutar diversas tareas o programas. Estos programas pueden ir desde un procesador de textos, a un programa para la animación de gráficos tridimensionales o distintos tipos de juegos. Para su correcto funcionamiento deben ser además capaces de acceder a los recursos de que dispone el ordenador, como por ejemplo escribir o leer datos en un disco duro, mostrar un gráfico por pantalla, etc. Es evidente, que si cada programa actuase de una forma independiente, existirían graves problemas y conflictos, puesto que, por ejemplo, tendrían libertad para escribir sus datos sobre los de otro, etc.

Para solucionar este tipo de problemas se desarrollaron los *Sistemas Operativos*, los cuales aportan unos mecanismos y reglas básicas de funcionamiento, de forma que los programas puedan acceder a los recursos del ordenador de una forma adecuada. Aunque ésta fue la funcionalidad inicial de los sistemas operativos, con el tiempo se han añadido otras muchas, como la ejecución de programas, el control de la memoria del ordenador, la creación y control de interfaces gráficas de usuario, etc.

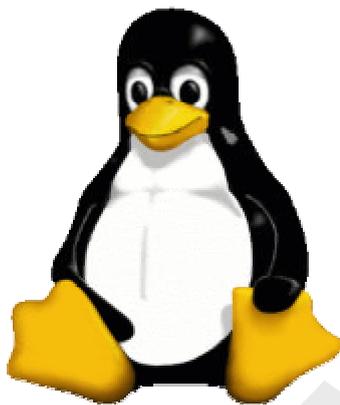


Figura 1

En la actualidad existen una gran cantidad de sistemas operativos dependiendo del tipo de ordenador en el que se va a ejecutar. Por ejemplo para los *PC* uno de los sistemas operativos más difundidos es *Microsoft Windows*, tanto en las versiones *95*, *98*, *2000* y *NT 4.0*. Otros posibles sistemas operativos para este tipo de ordenadores son *Solaris*, *OS/2*, *BeOS*, *Microsoft DOS*, o uno de los sistemas operativos más poderosos y en rápida expansión para *PC*, *LINUX*.

### 1.1 ¿Qué es *LINUX*?

*Linux* es un sistema operativo gratuito y de libre distribución inspirado en el sistema *Unix*, escrito por *Linus Torvalds* con la ayuda de miles de programadores en *Internet*. *Unix* es un sistema operativo desarrollado en 1970, una de cuyas mayores ventajas es que es fácilmente portable a diferentes tipos de ordenadores, por lo que existen versiones de *Unix* para casi todos los tipos de ordenadores, desde *PC* y *Mac* hasta estaciones de trabajo y superordenadores. Al contrario que otros sistemas operativos, como por ejemplo *MacOS* (Sistema operativo de los *Apple Macintosh*), *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. Pese a todo la enorme flexibilidad de *Linux* y su gran estabilidad ( y el bajo coste) han hecho de este sistema operativo una opción muy a tener en cuenta por aquellos usuarios que se dediquen a trabajar a través de redes, naveguen por *Internet*, o se dediquen a la programación. Además el futuro de *Linux* es brillante y cada vez más y más gente y más 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.

## 1.2 ¿Qué son las distribuciones?

Una de los primeros conceptos que aparecen al iniciarse en *Linux* es el concepto de *distribución*. Una *distribución* es un agrupamiento del núcleo del sistema operativo *Linux* (la parte desarrollada por *Linus Torvalds*) 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*.

A lo largo de este manual se considerará la *distribución* de *Linux* más extendida en la actualidad: *RedHat 6.0*. Aunque la mayor parte de la información debe ser válida para el resto de las distribuciones, existen determinadas opciones que están sujetas a cambio como el sistema de instalación del sistema operativo.

## 1.3 Entorno de trabajo: el shell y X Window

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* (no *X Windows*), el cual es el encargado de controlar y manejar la interfaz de usuario. Como se verá más adelante *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.

## 1.4 Usuarios y grupos

*Linux* es un sistema operativo *multitarea* y *multiusuario*. Esto quiere decir que es capaz de ejecutar varios programas (o tareas) de forma simultánea y albergar a varios usuarios de forma simultánea. Por lo tanto, todos los usuarios de *Linux* deben tener una cuenta de usuario en el sistema que establezca los privilegios del mismo. A su vez *Linux* organiza a los usuarios en *grupos* de forma que se puedan establecer privilegios a un determinado grupo de trabajo, para el acceso a determinados archivos o servicios del sistema.

## 2. INSTALACIÓN

Al contrario que lo que ocurre con *Microsoft Windows* la instalación de *Linux* no es un proceso sencillo, puesto que *Linux* permite el control y la personalización de una cantidad mayor de parámetros y opciones. Pese a todo se están realizando grandes progresos buscando que la instalación de *Linux* sea un proceso lo menos traumático posible, dependiendo la sencillez de la misma de la distribución que se emplee. Por el momento la distribución más sencilla de instalar es *Red Hat* y aquellas que derivan de esta (*Linux Mandrake*, ...).

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. El problema fundamental en todos los casos es que la mayor parte de esta información (aunque no toda) se encuentra en inglés. La distribución de *RedHat* incluye un manual de instalación de *Linux* en *HTML* en la que se explica con todo lujo de detalles cómo se realiza ésta y que es conveniente leer y comprender antes de proceder a realizar la instalación.

Uno 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. Como referencia se indica a continuación 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

Esto en general impide que se puedan instalar varios sistemas operativos *mezclados* en un mismo disco duro. Para solucionar este problema existen las llamadas *particiones* con las que se divide un determinado disco duro de forma que pueda contener ambos sistemas de archivos. A todos los efectos realizar una partición es equivalente a que el disco duro se divida en dos (aunque por supuesto no se divide de una forma física).

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 (un fichero muy grande), de las distribuciones *Corel Linux* y *Linux Mandrake 6.1*, o directamente en el sistema de archivos de *Windows* (*WinLinux 2000*). 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*, por lo que suele ser necesario realizar una partición del disco duro (Habitualmente *Windows* se apropia de la totalidad). Hasta épocas recientes esta división suponía la pérdida irremediable e inevitable de toda la información que contuviese el disco duro.

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**. (Mucho cuidado en no borrar la partición donde resida **Windows**, puesto que en ese caso se perderá **TODA** la información de forma permanente).

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 el PC. Los problemas surgen si hay que reinstalar **Windows** de nuevo, puesto que éste sistema operativo asume el control del 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 una de las formas más sencillas 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**.

Por último existe otro concepto fundamental a la hora de instalar y usar **Linux** que es el de **Super Usuario** o usuario **root**. Este usuario es el administrador del sistema y se crea durante la instalación. Como administrador que es puede acceder y modificar (así como destruir) toda la información del sistema, por lo que hay que evitar en la medida de lo posible trabajar como usuario **root**.

### 3. INICIACIÓN A *LINUX*

Existen tres formas de acceder a un sistema *Linux*:

- A través de una consola de texto, el usuario se conecta directamente al ordenador que tiene instalado *Linux* y accede mediante un sistema no gráfico.
- Desde un gestor de sesiones gráfico (*X Window*), el usuario se conecta directamente al ordenador que tiene instalado *Linux* y accede al sistema mediante un programa gráfico.
- Desde un ordenador remoto mediante *telnet* o *secure shell*

#### 3.1 Entrada y salida del sistema

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

**Login:** (Se teclea el nombre del usuario)

**Password:** (Se teclea la contraseña, que no se ve en la pantalla)

Por motivos de seguridad la contraseña debe cumplir 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. Existen diversas formas para terminar la sesión de trabajo en *Linux*, dependiendo de si estamos en modo gráfico o de texto.

En modo texto:

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

La salida de *X Window* depende del gestor de ventanas que se esté ejecutando y se explicará más adelante.

## 4. ESTRUCTURA DEL SISTEMA DE ARCHIVOS DE LINUX

### 4.1 Archivos: Tipos

La base del sistema de archivos de *Linux*, es obviamente el archivo, que no es otra cosa que la estructura empleada por el sistema operativo para almacenar información en un dispositivo físico como un disco duro, un disquete, un CD-ROM o un DVD. Como es natural un archivo puede contener cualquier tipo de información, desde una imagen en formato PNG o JPEG a un texto o una página WEB en formato HTML, ... El sistema de archivos es la estructura que permite que *Linux* maneje los archivos que contiene.

Todos los archivos de *Linux* tienen un nombre, el cual debe cumplir unas 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, y por lo tanto no es lo mismo *carta.txt* que *Carta.txt* ó *carta.Txt*

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*, *.htm* (o *.html*), *.png* y *.jpg* (o *.jpeg*) respectivamente.

Pese a esto *Linux* sólo distingue tres tipos de archivos:

- *Archivos* o ficheros ordinarios, son los mencionados anteriormente.
- *Directorios* (o carpetas), es un archivo especial que agrupa otros ficheros de una forma estructurada.
- *Archivos especiales*, son la base sobre la que se asienta *Linux*, puesto que representan los dispositivos conectados a un ordenador, como puede ser una impresora. De esta forma introducir información en ese archivo equivale a enviar información a la impresora. Para el usuario estos dispositivos tienen el mismo aspecto y uso que los archivos ordinarios.

### 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 trabajen sobre un mismo fichero, puesto que permiten compartir el fichero pero centralizan las modificaciones.

Como ejemplo se puede suponer la existencia de un fichero llamado *balance.1999.txt*, al que se crea un enlace simbólico *balance.txt*. Cualquier acceso a *balance.txt* es traducido por el sistema de forma que se accede al contenido de *balance.1999.txt*.

### 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, estando compuesto cada nudo por un directorio o carpeta, que contiene otros directorios o archivos. En *Windows* cada unidad de disco se identifica como una carpeta básica que sirve de raíz a otras, y cuyo nombre es especial *a;*, *c;*, *d;* etc. En los sistemas *Unix*, y por lo tanto en *Linux*, existe una única raíz llamada */* de la que cuelgan todos los ficheros y directorios, y que es independiente de qué dispositivos estén conectados al ordenador.

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 */*. Supongamos la estructura de archivos de la Figura 2.

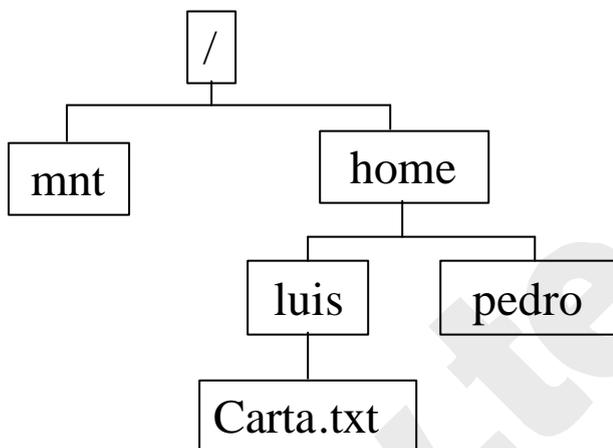


Figura 2

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*

Para complicar aun más las cosas, todos los directorios contienen dos directorios especiales:

- El directorio actual, representado por el punto *.*
  - El directorio padre representado por dos puntos *..*
- Estando en el directorio */home/pedro* se puede acceder a *Carta.txt* con */home/luis/Carta.txt* (path absoluto) o bien *../luis/Carta.txt* (path relativo). En *luis* como *./Carta.txt* o simplemente *Carta.txt*.

### 4.4 Estructura del sistema de archivos de Linux

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:*.

Todos 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 sistema **Unix**.

- / Raíz del sistema de archivos.
- /dev** Contiene ficheros del sistema representando los dispositivos que estén físicamente instalados en el ordenador.
- /etc** Este directorio esta reservado para los ficheros de configuración del sistema. En este directorio no debe aparecer ningún fichero binario (programas). Bajo este deben aparecer otros dos subdirectorios:
  - /etc/X11** Ficheros de configuración de X **Window**
  - /etc/skel** Ficheros de configuración básica que son copiados al directorio del usuario cuando se crea uno nuevo.
- /lib** Contiene las librerías necesarias para que se ejecuten los programas que residen en **/bin** (no las librerías de los programas de los usuarios).
- /proc** Contiene ficheros especiales que o bien reciben o envían información al **kernel** del sistema (Se recomienda no modificar el contenido de este directorio y sus ficheros).
- /sbin** Contiene programas que son únicamente accesibles al superusuario o **root**.
- /usr** Este es uno de los directorios más importantes del sistema puesto que contiene los programas de uso común para todos los usuarios. Su estructura suele ser similar a la siguiente:
  - /usr/X11R6** Contiene los programas para ejecutar X **Window**.
  - /usr/bin** Programas de uso general, lo que incluye el compilador de C/C++.
  - /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.

<i>/usr/lib</i>	Librerías generales de los programas.
<i>/usr/man</i>	Manuales accesibles con el comando <i>man</i> (ver más adelante).
<i>/usr/sbin</i>	Programas de administración del sistema.
<i>/usr/src</i>	Código fuente de programas.

Existen además de los anteriores otros directorios que se suelen localizar en el directorio */usr*, como por ejemplo las carpetas de los programas que se instalen en el sistema.

*/var* Este directorio contiene información temporal de los programas (lo cual no implica que se pueda borrar su contenido, de hecho, **¡no se debe hacer!**)

#### 4.5 Acceso a los diferentes sistemas de archivos

Como se ha visto anteriormente el sistema de archivos de *Linux* sólo tiene una raíz y su estructura es independiente de los dispositivos de almacenamiento existentes. Esto implica que 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”, que se verá más adelante. Cuando se ha terminado de trabajar con un determinado dispositivo hay que “desmontarlo” (**¡No físicamente!**).

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).

Para complicar más las cosas sólo el administrador o *root* tiene permiso para montar y desmontar un sistema de archivos (por motivos de seguridad), aunque esto puede ser arreglado.

## 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 el 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**), ó 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.

## 5. X WINDOW

*X Window* es el entorno gráfico habitual de los sistemas *Unix*. El sistema *X Window* se compone de dos partes principales el *servidor X* y el programa para la gestión de las ventanas. El *servidor X* es el programa que se encarga realmente de dibujar en la pantalla. Por el contrario el gestor de ventanas como su nombre indica es el encargado de crear las ventanas y gestionar su apariencia. Debido a este modelo, la apariencia de las aplicaciones varía según se use uno u otro gestor de ventanas, entre los que destacan por su sencillez de uso los entornos *GNOME* y *KDE*.

Al instalar *Linux* el sistema puede preguntar si se desea arrancar *Linux* en modo texto o en modo gráfico. Si se ha seleccionado esta última opción *Linux* arrancará directamente *X Window*, en caso contrario en la línea de comandos hay que escribir *startx* con lo cual se arranca el modo gráfico. Por defecto esto arranca el entorno gráfico *GNOME* (en distribuciones que no sean *Red Hat 6.0* se puede arrancar por defecto otro entorno como *KDE* o *AfterStep*)

## 6. GNOME

Cuando *Linux* arranca en modo gráfico aparece una ventana similar a la siguiente:



Figura 3

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

**Nota:** tanto *Linux*, como *UNIX*, distinguen entre letras mayúsculas y minúsculas. Por lo que `root` no es lo mismo que `Root` o `ROOT`.

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. Nota: **NUNCA** se debe apagar directamente el ordenador siempre hay que apagarlo empleando la opción *Halt* anterior (tras salir del entorno con *logout*) o empleando el comando *halt* en el caso de trabajar en modo texto.

Tras introducir el nombre del usuario y el password aparecerá una pantalla similar a la siguiente:

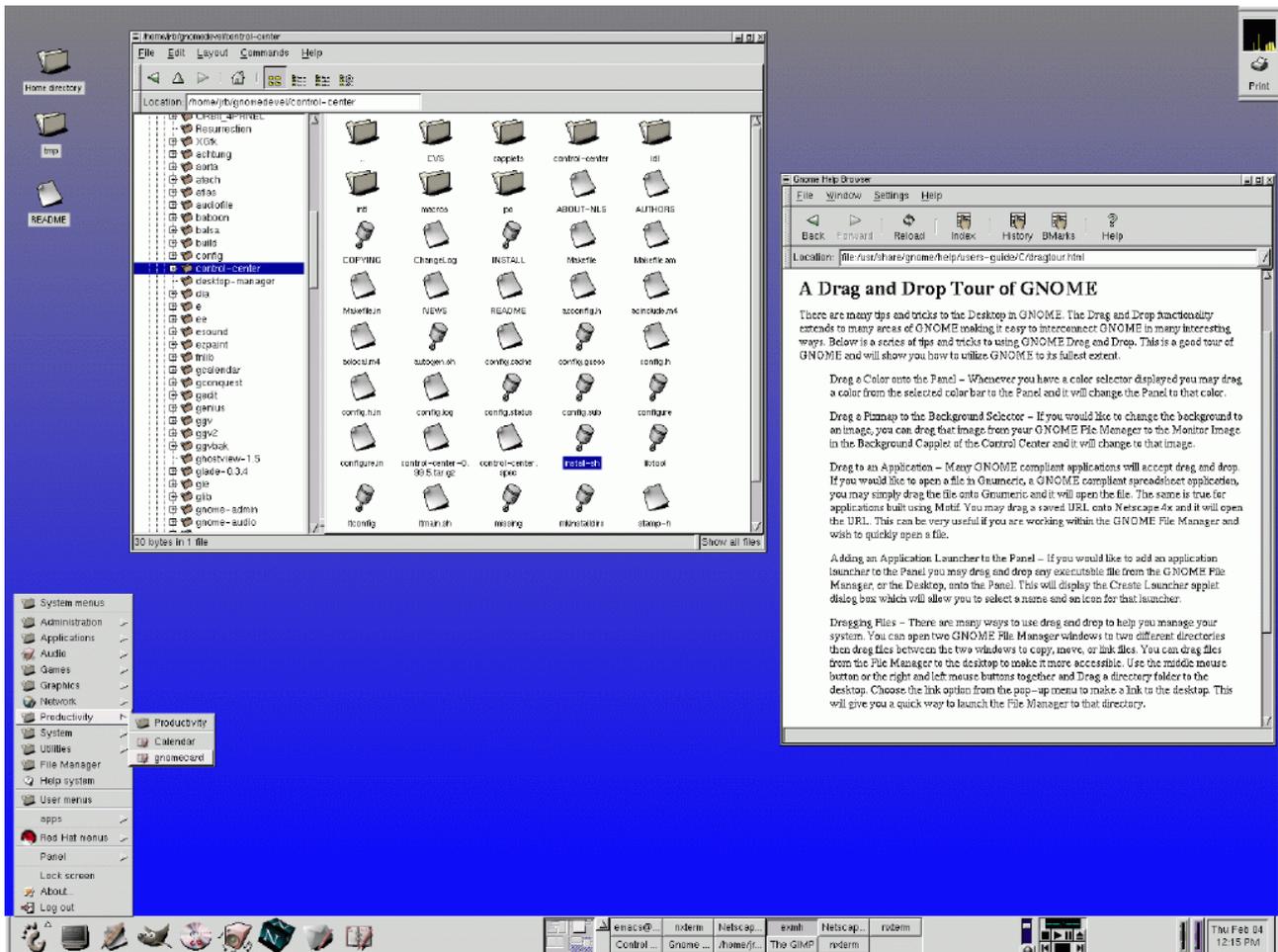


Figura 4

Como se puede observar en la figura anterior este entorno es muy similar a otros como **Windows**, **OS/2** o **Macintosh**. Al igual que estos entornos **GNOME** está diseñado para ser empleado con el ratón, e incluye elementos comunes con estos entornos como iconos, menús, etc. Al igual que **Windows** incluye en la parte inferior una barra, el Panel de **GNOME** (**GNOME Panel**), en la cual se encuentran accesos directos a determinados programas de uso común.

En **GNOME** para acceder a las distintas aplicaciones hay que seleccionar el botón en forma de huella de pie como se aprecia en la figura adjunta, lo cual despliega un menú. Como se puede comprobar existen una gran cantidad de programas disponibles para el usuario, así como la ayuda del sistema la cual incluye un tutorial sobre su uso.

Para salir de **GNOME** hay que seleccionar la opción **Log Out** que aparece en la parte inferior del menú.

Si se ha entrado como usuario **root** por primera vez tras instalar **Linux** el primer paso a seguir es crear una nueva cuenta de usuario que permita acceder al sistema como un usuario normal sin los privilegios del superusuario, de forma que se impida la modificación del sistema de forma inadecuada.

Si se está familiarizado con *MS-DOS* o con versiones de *Windows* como *Windows 98* y *Windows 95*, puede parecer un poco irracional tener que crear una nueva cuenta de usuario. Después de todo al entrar con *root*, se puede navegar por el sistema y usar programas por lo que puede parecer excesivo tener dos cuentas en el mismo ordenador.

Nada más lejos de la realidad:

*Linux* es un sistema operativo multitarea y multiusuario, lo que significa que puede ser empleado de forma segura por varios usuarios de forma simultánea, realizando todas las acciones que los usuarios deseen. Pero por motivos de seguridad sólo puede existir un único usuario *root*, capaz de cambiar la forma de trabajo del sistema.

Además actuar como superusuario puede conducir al desastre puesto que el sistema permite modificar todos sus parámetros, lo cual puede conducir a daños irreparables. Por este motivo cada usuario de *Linux* debe tener su cuenta de usuario, aun siendo los administradores del sistema, trabajando como *root* sólo en aquellos casos en los que se necesita realizar la administración del sistema, como por ejemplo al crear un nuevo usuario. Al entrar en el sistema como *root*, se puede ejecutar el programa *linuxconf*, que se encuentra en *System/linuxconf*. Este programa permite manipular una cantidad enorme de parámetros del sistema entre los que se encuentra la administración de usuarios. Este programa es un ejemplo de porqué actuar como superusuario es peligroso, *linuxconf* permite crear y borrar usuarios, cambiar contraseñas, etc.

La primera pantalla de *Linuxconf* es similar a la siguiente:

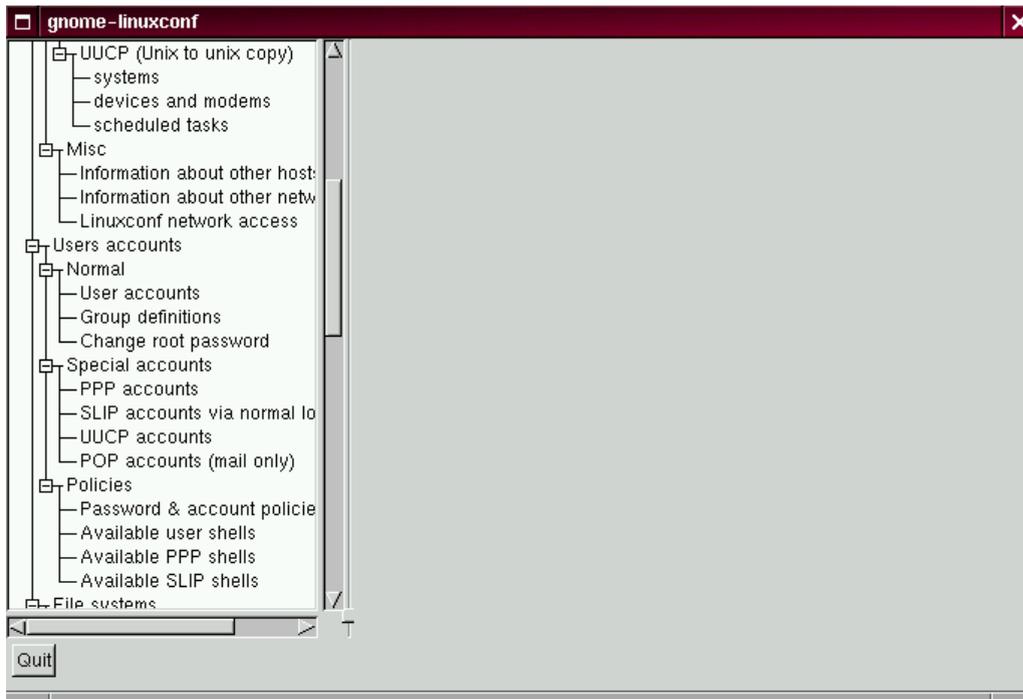


Figura 5

En donde en la lista se puede seleccionar *User accounts/Normal/User accounts*, tras lo que aparecerá un cuadro de diálogo como el siguiente:

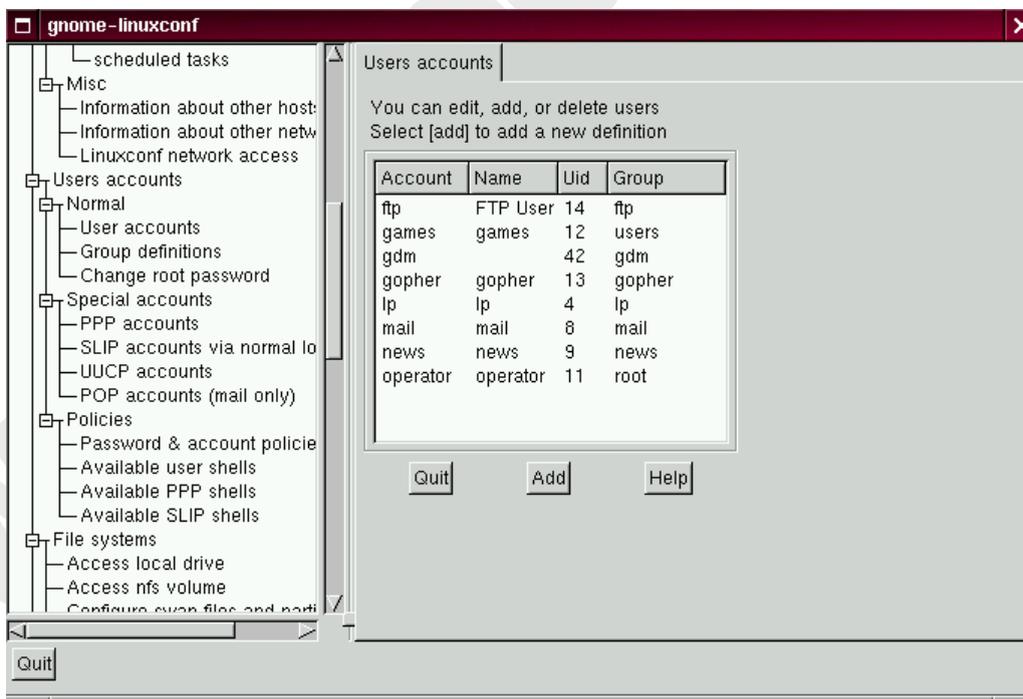


Figura 6

Tras seleccionar **Add** se muestra el siguiente cuadro de diálogo:

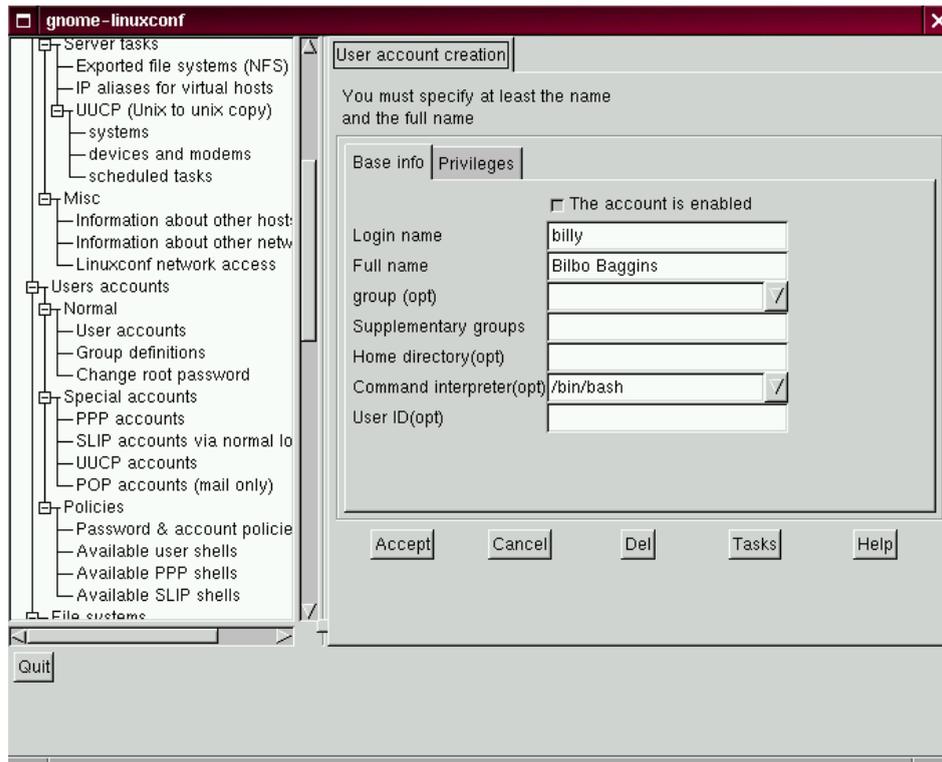


Figura 7

En el anterior cuadro de diálogo hay que especificar el nombre de usuario: por ejemplo billy (Hay que recordar que se distingue entre mayúsculas y minúsculas por los que *no* es lo mismo *billy* que *Billy*), el nombre completo del usuario *Bilbo Baggins*. Los demás parámetros son opcionales y no se comentarán en este momento. Por defecto el directorio en el que este usuario podrá escribir los ficheros será algo así como /home/billy. Tras aceptar el sistema pedirá el nuevo password del usuario.

### 6.1 Iniciación a GNOME

El uso de *GNOME* es muy parecido al de otros entornos gráficos. La pantalla inicial de *GNOME* se encuentra dividida en dos zonas principales como se puede apreciar en la siguiente figura:

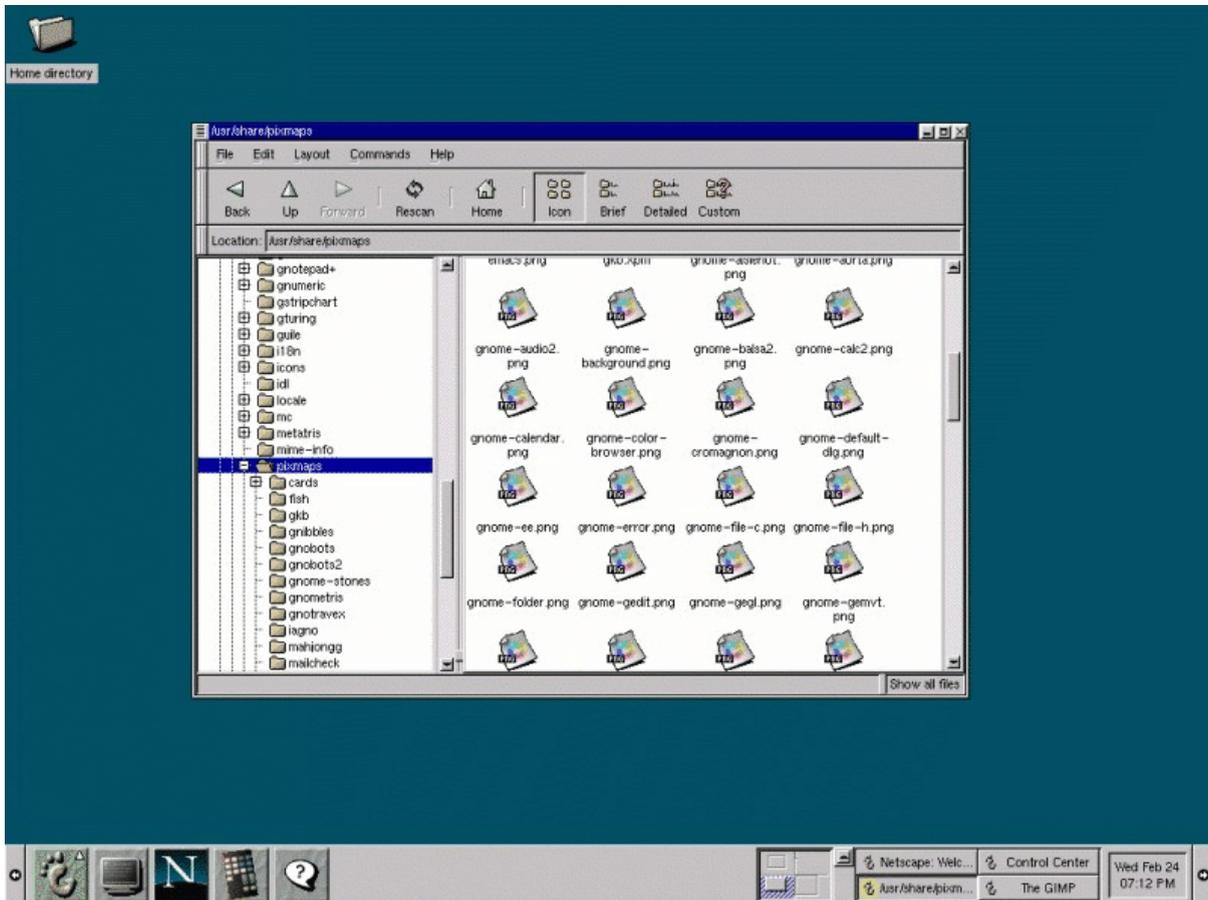


Figura 8

La parte superior en la que aparecen las ventanas de las aplicaciones y documentos del usuario recibe el nombre de *escritorio*, mientras que la parte inferior de la pantalla recibe el nombre de *panel de GNOME*:

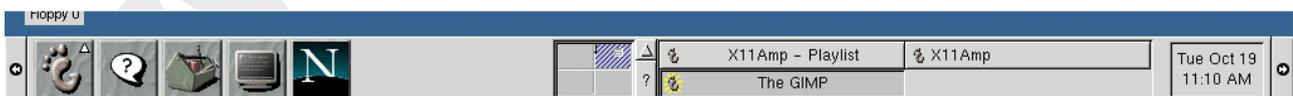


Figura 9

Este elemento está diseñado para contener la siguiente información:



- Contiene la *huella de pie* o *footprint*, al seleccionar este elementos aparece un menú similar al menú *Start* de *Windows 9x* con las aplicaciones más importantes instaladas.
- Las *applets* son enlaces a las aplicaciones de uso más frecuente como la consola, *Netscape*, la ayuda, etc.

- El acceso a los escritorios virtuales. Al contrario que en *Windows*, *X Window* permite organizar las ventanas en varios escritorios virtuales.
  - Al igual que *Windows* el panel de *GNOME* dispone de un área específica en la que aparecen los botones representativos de las ventanas.
  - En los dos extremos del panel aparecen dos botones con flechas con los que el usuario puede colapsar el panel de forma que se maximice el área útil del escritorio.

Clicando 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 todas las características del *applet*.

Seleccionado así mismo en alguna de las áreas libres del panel se despliega otro menú que permite configurar todo el panel y añadir y quitar *applets* del mismo. Este mismo menú puede ser accedido a través de la opción Panel del menú principal (*footprint*).

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

## 6.2 Aplicaciones auxiliares de GNOME

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

### 6.2.1 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 archivos.

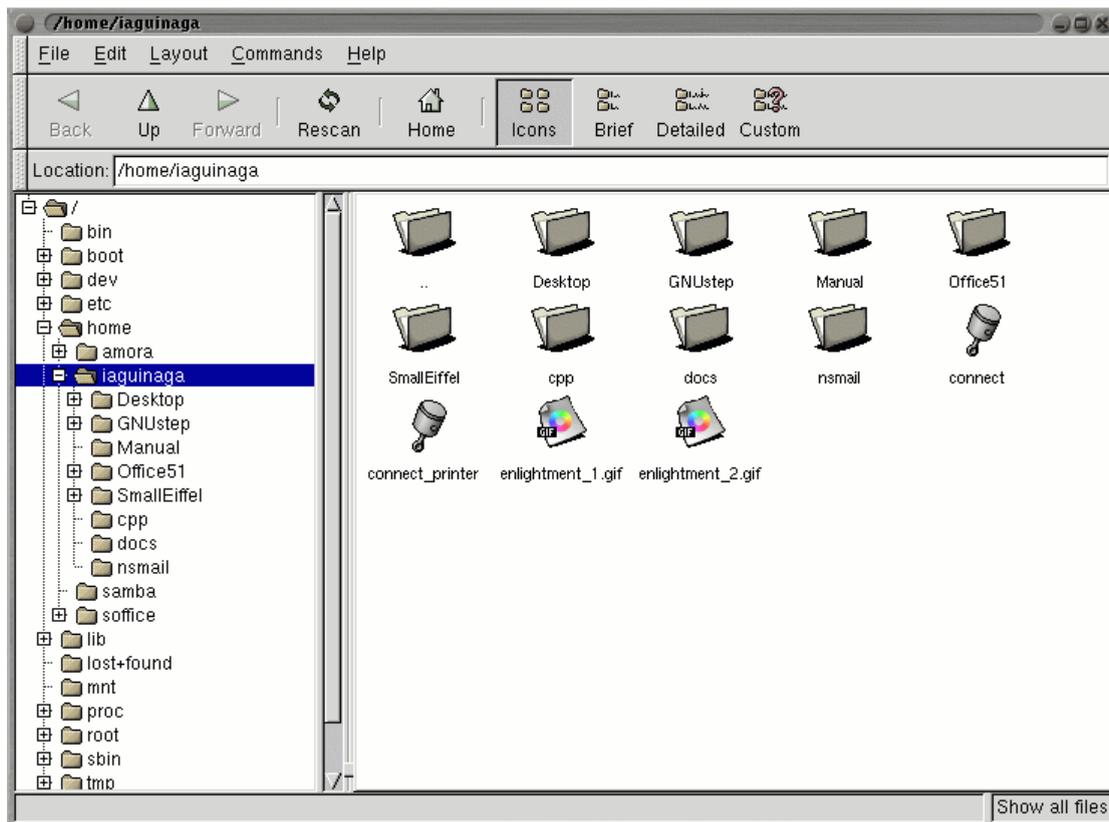


Figura 10

El *administrador de archivos* o *file manager* permite seleccionar archivos 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 clicando en los ficheros de los extremos.

Para mover un directorio, un archivo o una selección basta con mantener clicado el ratón sobre él y arrastrarlo a su nueva posición. Para copiar un directorio, un archivo o una selección se realiza también la operación anterior pero manteniendo pulsada la tecla **<Ctrl>**. Finalmente para eliminar directorios, ficheros o una selección hay que clicar sobre ellos con el botón derecho del ratón y aparecerá un menú en el que habrá que escoger la opción *delete*. Si se realiza un doble clic sobre un fichero se abrirá un cuadro de diálogo que preguntará el programa con el que se desea abrir ese fichero. Si se ha realizado un doble clic sobre un directorio se mostrarán los archivos y subdirectorios que contiene. En el caso de haberse realizado un doble clic 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 y que son utilizados por los programas o el sistema para guardar su configuración). El icono *Rescan* permite actualizar la lista de ficheros tras realizar un cambio.

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

- **Fichero?.txt** indicaría cualquier fichero cuyo nombre empiece por **Fichero** y termine en **.txt** pudiendo existir cualquier carácter entre ambos, pero sólo uno, por ejemplo: **Fichero1.txt**, **Fichero2.txt**, **FicheroZ.txt**, etc.
- **Fichero\*.txt** indicaría cualquier fichero cuyo nombre empiece por **Fichero** y termine en **.txt** pudiendo haber entre ambos cualquier conjunto de caracteres, por ejemplo: **Fichero1.txt**, **Fichero\_de\_texto.txt**, etc.

### 6.2.2 GNOME Search Tool

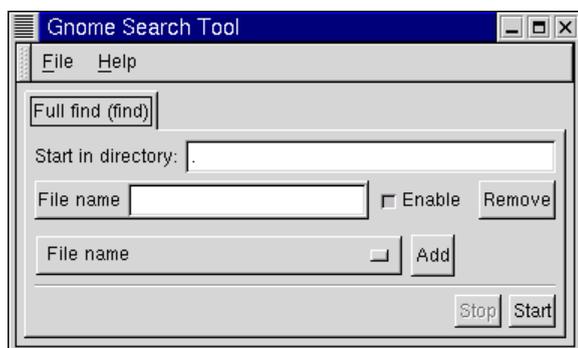


Figura 11

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** (activar) 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.

### 6.2.3 Color Xterm, GNOME Terminal y Regular Xterm

Se encuentran en el menú **Utilities** y sirve para abrir una consola de **Linux** en la cual introducir comandos. La utilidad y el funcionamiento de las consolas se explicará más adelante.

### 6.2.4 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 la amplificación, la ganancia, ... También da la opción de silenciar (**mute**) las fuentes de sonido que se desee.

**CD Player** : permite la reproducción de CDs de música.

### 6.2.5 Otras aplicaciones

**GQ View:** Este programa 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:** Este pequeño programa se encuentra en el menú **Utilities** e indica el espacio ocupado de los dispositivos que están siendo utilizados (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 (1/0, ...) mostrará el símbolo e. Este símbolo se podrá borrar mediante la tecla AC.

**GnomeCard:** Este programa se encuentra en el menú **Applications**. Sirve para gestionar direcciones (tanto electrónicas como no electrónicas) y para crear tarjetas.

**Calendar:** Este programa 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 (Tipo de CPU, Distribución de **Linux** y su versión, ...). Si se hace clic en **Detailed Information** suministra información extra acerca de la CPU, la memoria y el disco o discos duros del ordenador.

#### **Change Password**

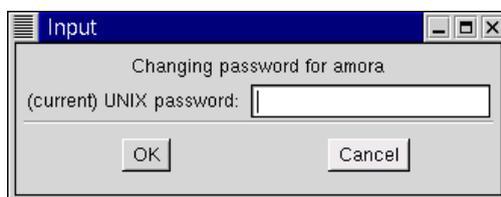


Figura 12

Este programa se encuentra en el menú **System** y su finalidad es obvia. 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**. Permite monitorizar los sistemas de archivos.

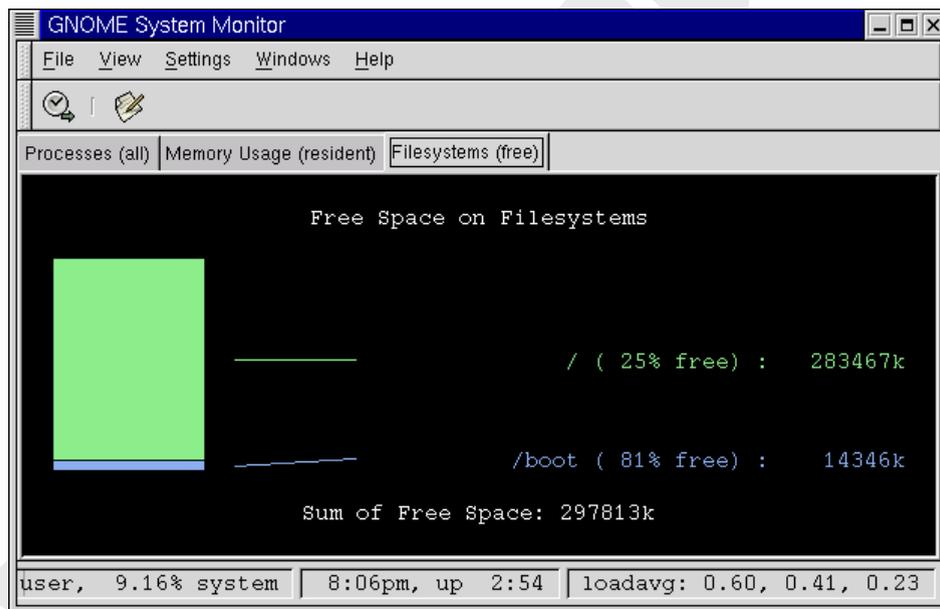


Figura 13

También permite monitorizar la memoria utilizada por diversos programas o procesos que se están ejecutando en ese instante.

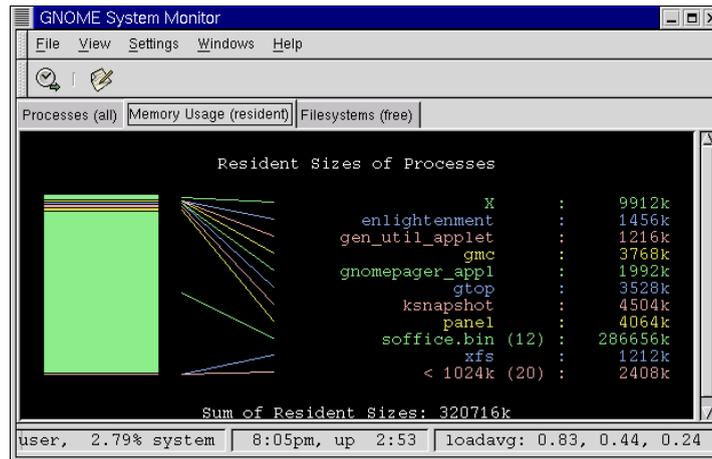


Figura 14

Por último permite visualizar todos los procesos que se están ejecutando en cada instante.

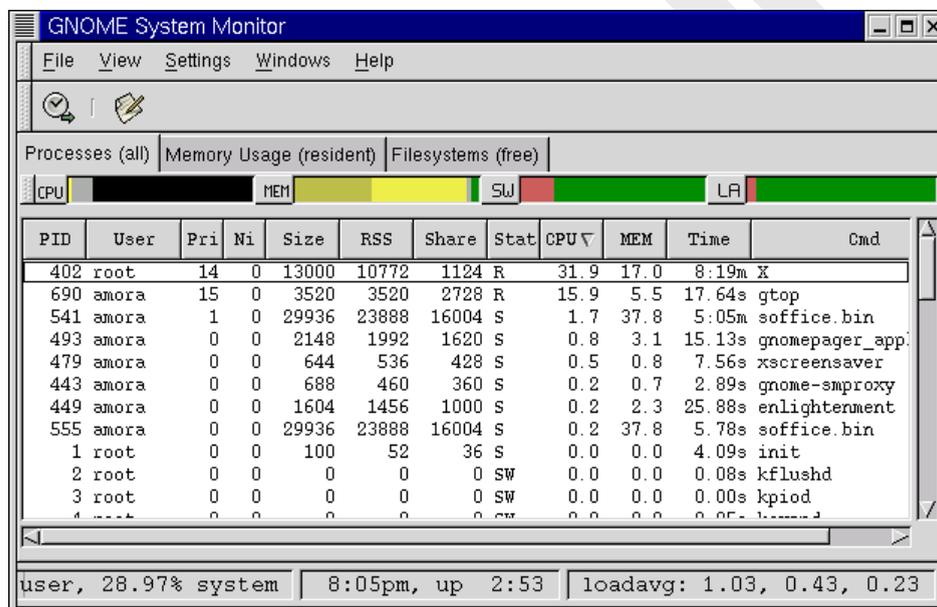


Figura 15

Además de la memoria utilizada por cada proceso indica el tanto por ciento de CPU que utiliza y el propietario del proceso (el usuario que lo ejecutó). Si se hace clic dos veces sobre cualquiera de los procesos se muestra información detallada de ese proceso mientras que si se hace clic con el botón derecho del ratón el usuario puede decidir entre ver los detalles del proceso, reiniciarlo, o enviar diversas señales (*Signal Kill*, *Signal Term*, ...). Si alguno de los procesos se ha quedado bloqueado se puede arrancar este programa y eliminar el proceso en cuestión enviándole una señal *Kill*, *Term*, ... El que se envíe una señal u otra dependerá del grado de bloqueo del programa. También existen más señales disponibles que obligan al proceso a realizar diversas tareas.

**gedit:** Este programa 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 (**Linewrap**). 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 (**Spell Check**), y el **plugging Email**. Este último **plugging** 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 cuales son los programas con los que pasa mas tiempo. Si el usuario cobra por hora de trabajo con un determinado programa esta utilidad le permite medir el tiempo de trabajo. 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. En las fórmulas matemáticas intervienen celdas con valores numéricos previamente rellenados y funciones que aporta el programa tales como SUM, AVERAGE, ... Por ejemplo: =SUM(A1:A4)+B1 indicaría que la celda que contiene esa formula contendrá la suma de los valores contenidos en las celdas A1, A2, A3, A4 y B1.

Para editar el contenido de una celda basta con clicar sobre ella con el ratón. Si se mantiene clicado el ratón sobre la esquina inferior derecha de una celda y se arrastra el ratón el contenido de esta celda se copia en las adyacentes por filas o columnas según halla sido el movimiento del ratón. Si se realiza la operación anterior sobre una selección el programa considera la selección como un serie y calcula los siguientes elementos de la serie (llamado copiado inteligente). Por ejemplo si dos celdas consecutivas tuviesen los valores 1 y 2 y utilizando el copiado inteligente en las siguientes celdas se continuaría la serie, 3, 4, ... Para seleccionar una fila entera o una columna basta con clicar sobre el número de la izquierda que indica la fila o la letra superior que indica la columna. Para seleccionar toda la tabla hay que clicar en el cuadro superior izquierdo donde se cruzan la barra que contiene los números de las filas y la barra que contiene las letras que identifican cada columna.

### 6.3 Configuración de GNOME

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

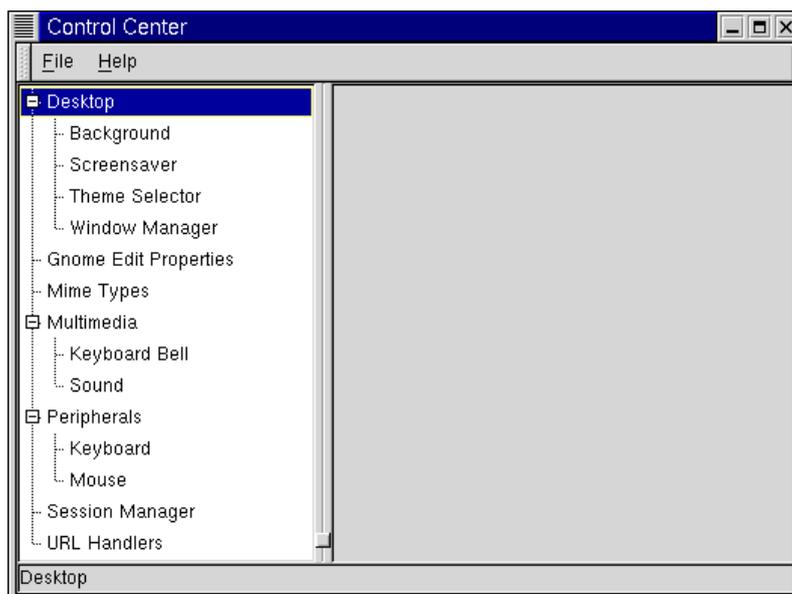


Figura 16

Esta aplicación 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. Así mismo, todas las opciones de este programa como **Desktop/Background**, **Mime Types**, ... son accesibles directamente desde el menú **Settings**.

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.

**Window Manager** permite configurar el programa que se encarga de dibujar las ventanas. En principio se puede utilizar cualquier **gestor de ventanas** pero por ahora (debido a que **GNOME** es muy reciente) el

único gestor que es compatible con **GNOME** y aprovecha todas sus características es **Enlightenment**. Esta aplicación tiene su propio programa de configuración, el cual permite cambiar incluso el aspecto de las ventanas. Ese programa de configuración se explicará más adelante.

En el apartado **GNOME 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 (entrar en el sistema, desplegar el menú principal, ...).

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 (**right handed**) o una persona zurda (**left handed**). En el último caso el funcionamiento de los botones del ratón será justamente el inverso (el botón derecho del ratón actuará como el izquierdo y viceversa).

En **Session Manager** el usuario puede indicarle al sistema que le pida confirmación antes de acabar la sesión (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 qué programas se arrancarán automáticamente cada vez que el usuario comience una nueva sesión (cuando el usuario haga **login**).

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 la siguiente:

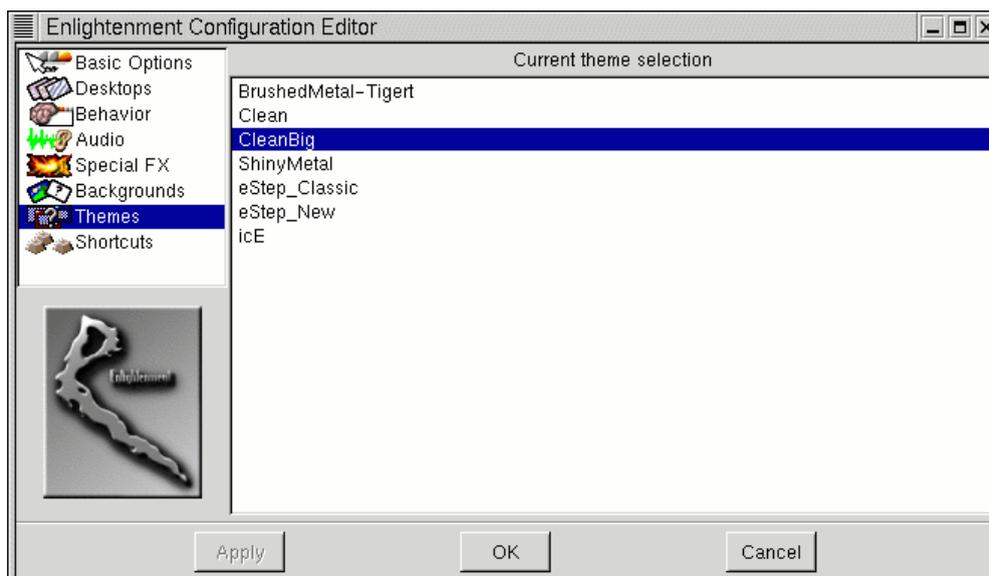


Figura 17

En el apartado **Basic Options** el usuario puede seleccionar el modo de arrastre y de cambio de tamaño de las ventanas (La forma de representación gráfica de esas dos operaciones). También permite indicar al sistema que la ventana activa sea siempre aquella sobre la que se encuentra el cursor o la última ventana sobre la que se ha clicado.

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 esta desactivado para pasar de un **Desktop** a otro habrá que utilizar el **Applet Gnome Pager** que se encuentra en el panel de **GNOME** o **ShortCuts** (combinaciones de teclas concretas).

En el apartado **Behaviours** el usuario puede modificar el comportamiento de las ventanas. El programa de configuración le permite elegir cuándo y cómo desea que una ventana se active así como configurar los **tooltips** (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 **Special FX** permite al usuario las animaciones que se mostrarán cuando se cambia de escritorio virtual, cuando se activa o se arrastra una ventana, ...

El apartado **Background** permite elegir el fondo del escritorio. El fondo puede ser simplemente un color sólido seleccionado por el usuario o una imagen también seleccionada por el usuario. Se puede crear una lista de fondos preferidos. Así mismo cada fondo puede ser editado indicando (si se trata de una imagen) que se muestre un mosaico con la imagen, que se ajuste la anchura o altura de la imagen a la anchura o altura de la pantalla o que se mantengan las proporciones de la imagen.

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. Por ejemplo pulsando <Alt>+<Shift>+ la tecla de cursor arriba, abajo, izquierda, derecha se cambia de un escritorio al contiguo según la dirección indicada por la tecla de cursor pulsada.

Hay varias formas de configurar el panel de **GNOME**. Clicando sobre cualquier elemento de la barra del panel con el botón derecho del ratón aparecerá un menú contextual, con la opciones de eliminar dicho elemento del panel de **GNOME**, cambiarlo de sitio o editar sus propiedades. Si se clica sobre una zona del panel que esté 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*, etc.

**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 (*GNOME Pager*),

Clicando con el botón derecho del ratón sobre una zona libre del panel de **GNOME** aparece el menú contextual de la Figura 18.

Con la opción *Add new panel* el usuario puede añadir mas 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 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.

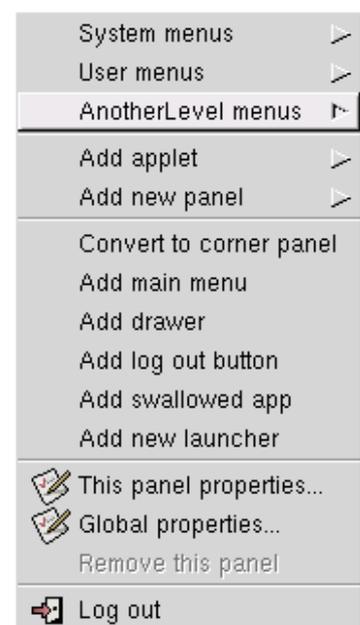


Figura 18



Figura 19

Con la opción **Add log out button** el usuario puede añadir al Panel un botón que de 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. Si se clicca sobre el cajón este se abrirá hacia arriba. Si una vez abierto el cajón se clicca sobre la "bandeja" abierta con el botón derecho del ratón, aparecerá un menú igual al que aparece cuando se realiza esa operación sobre una zona libre del Panel. A todos los efectos el **Drawer** se comporta de la misma manera que una panel de **GNOME**, por lo tanto en el se pueden introducir **Applets, launchers, ...**

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 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.

Con la opción **Global panel properties** también del menú de la página anterior se pueden configurar los **tooltips**, las animaciones de los elementos del panel, los fondos que se utilizarán para los botones cuando estén presionados y cuando no lo estén, ...

El último elemento de **GNOME** que falta por configurar es el menú principal. Para configurar el menú principal hay que utilizar el programa **Menu Editor** que se encuentra en el menú **/Settings/Menu Editor** del menú principal.

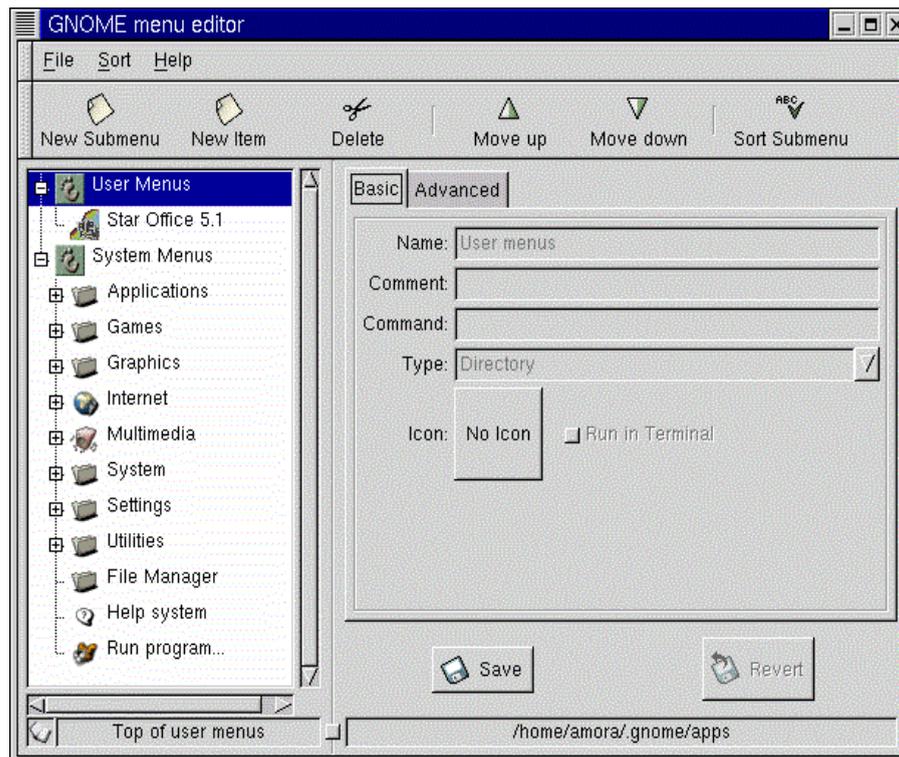


Figura 20

Mediante los iconos *New Submenu* y *New Item* 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 (*root*) pero para modificar el *User menu* no hace falta. Una vez modificados los menús el usuario deberá guardar los cambios (*Save*) y salir del programa. Antes de guardar los cambios el usuario puede pedir al programa que ordene alfabéticamente los elementos de los menús.

## 7. KDE

### 7.1 Partes de la pantalla

**KDE** es uno de los entornos gráficos más populares de **Linux** puesto que une una gran facilidad de uso a un entorno bonito y agradable. Al arrancar **KDE** aparece el escritorio en el que se pueden encontrar elementos similares a los de otros entornos:

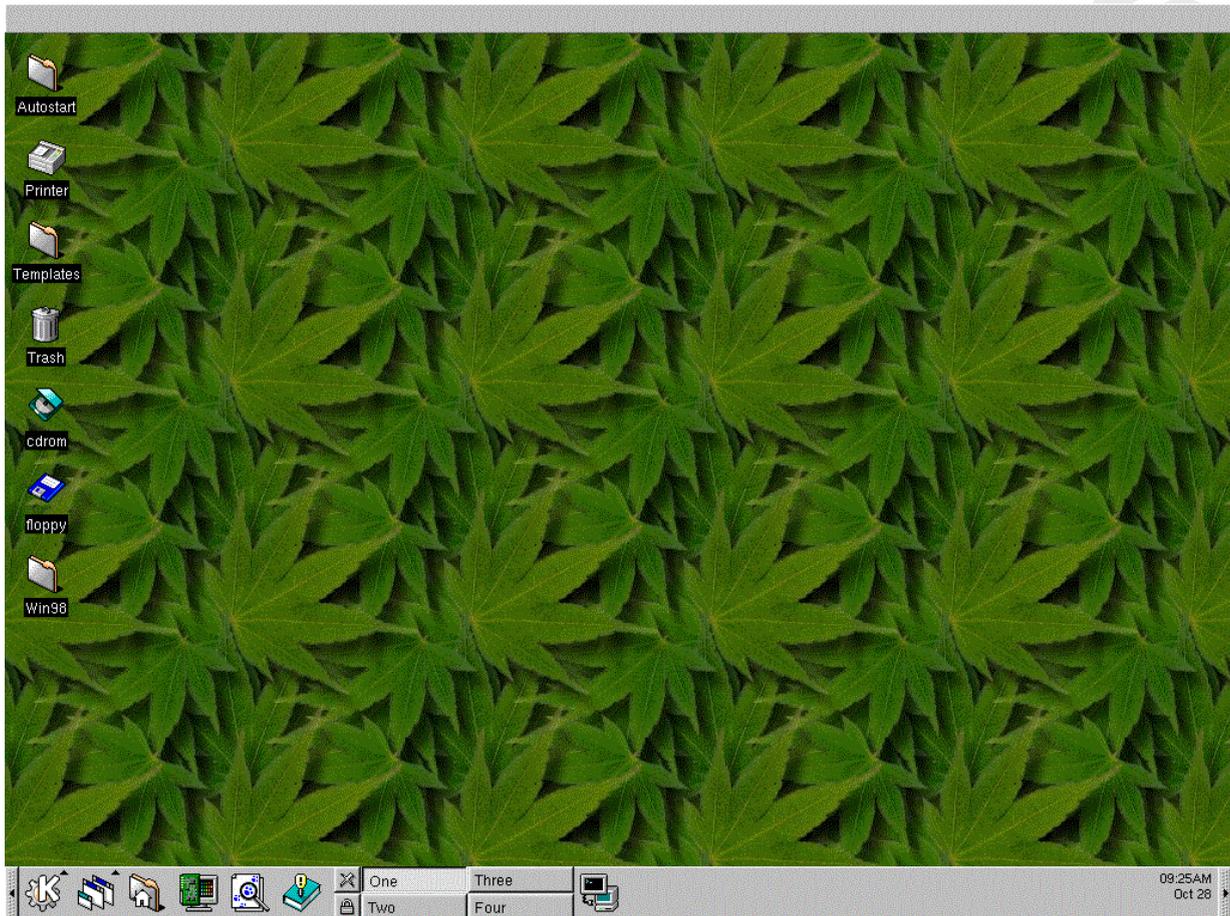


Figura 21

Por defecto la pantalla de **KDE** se divide en tres partes fundamentales:

- Panel de **KDE**
- Escritorio
- Panel de ventanas

El panel de **KDE** contiene accesos directos a las aplicaciones más empleadas así como dos menús.





El equivalente al menú *Start* de *Windows*, esto es el menú a través 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.



Junto a éste aparece un segundo menú del *KDE*, en el menú de ventanas se puede 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 escoger uno de los cuatro botones que aparecen en el panel.

Justo encima del panel de *KDE*, aparece el escritorio, al igual que en *Windows* este elemento contiene iconos que permiten acceder a los elementos más comunes como las unidades de disco o la papelera. Por último en la parte superior del escritorio aparece otra barra, en la que aparecerán botones por cada ventana que se cree.

Las ventanas en el *KDE* tienen un aspecto similar al de las ventanas de *Windows* (al menos con el aspecto básico), pudiendo distinguir en ellas diversas partes:

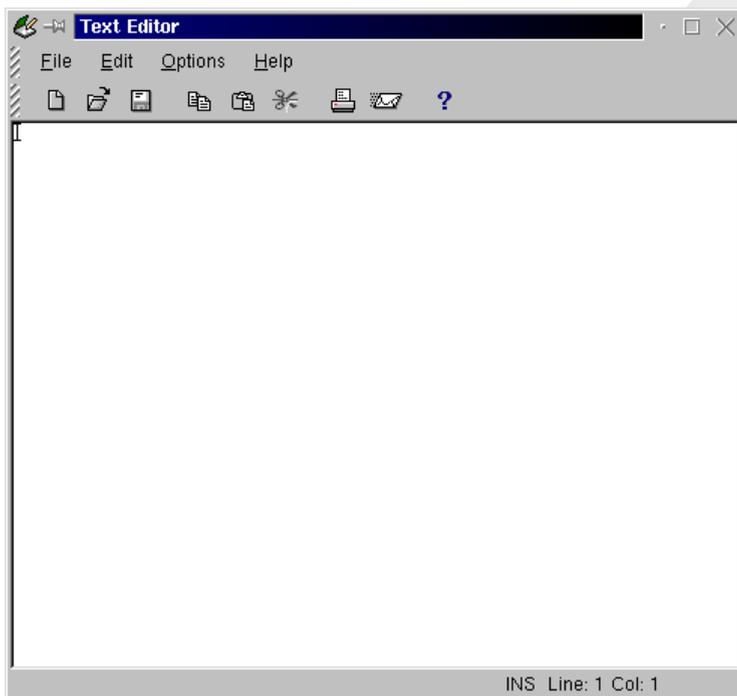


Figura 22

En la parte superior izquierda, aparece el icono de la aplicación, seleccionando el cual aparece un menú con las opciones básicas de manipulación de ventanas: minimizar, maximizar, cerrar; así como otras no tan habituales como enviar la ventana a otros escritorio. Junto a él y en la parte superior central de la ventana se encuentra la barra de títulos de la ventana. Finalmente en la parte superior derecha aparecen tres botones con las opciones de minimizar, maximizar y cerrar. Esta es la disposición por defecto pero como se verá más adelante esta disposición puede ser adaptada a los gustos del usuario de una forma muy sencilla. Por debajo de este elemento se extiende la barra de menús y de herramientas y el área de trabajo de la aplicación.

Al igual que en *Windows*, *KDE* permite cambiar el tamaño de una ventana sin más que acercar el ratón a un borde de la misma. En esta posición cambia el cursor, indicando en que dirección podemos modificar el tamaño de la ventana en esa posición. Si se hace clic sobre el borde y se arrastra cambiará el tamaño de la ventana.

## 7.2 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, buscarlos, abrirlos para ser editados etc. **KDE** incluye una herramienta, el *kfm*, que permite realizar todas estas operaciones de una forma sencilla.



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

Esto abre una ventana similar a la siguiente en la que se muestra el contenido de una determinada carpeta del sistema:

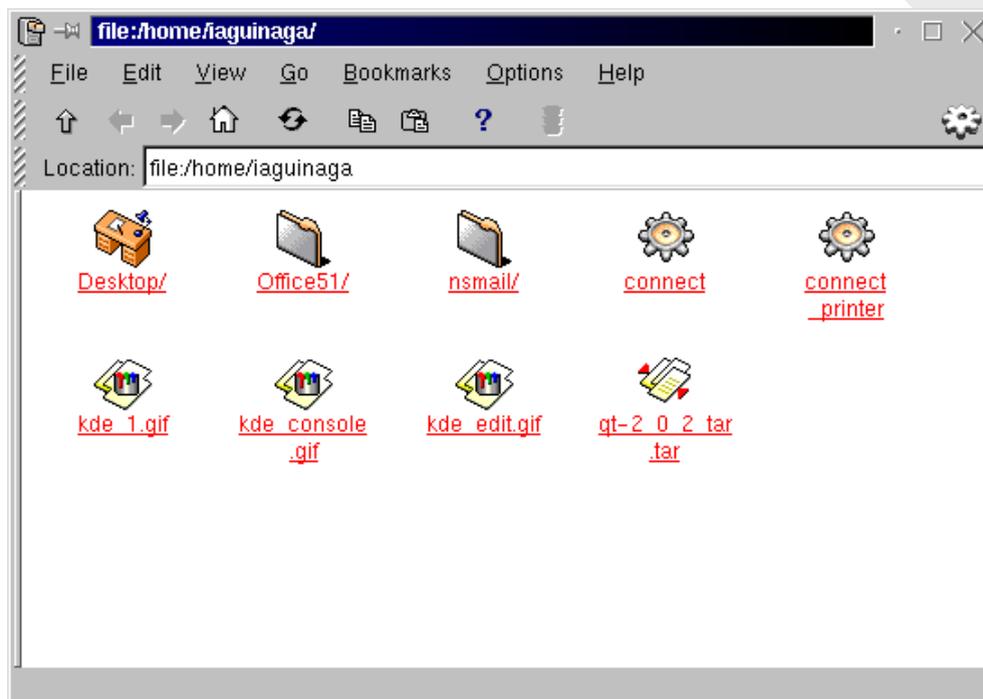


Figura 23

### 7.2.1 Navegar por la estructura de directorios y ver el contenido de los ficheros

El *kfm* está diseñado de tal forma que su uso y funcionamiento es muy similar al de un navegador de **Internet** como **Microsoft Internet Explorer** o **Netscape Navigator**, tanto es así, que este programa es capaz de abrir y mostrar ficheros **HTML** y puede ser configurado para acceder a **Internet**.



Figura 24

Al igual que estos programas contiene botones en la barra de herramientas para facilitar la navegación tanto por los directorios como por *Internet*:



Figura 25

Los botones que aparecen en esta barra, comenzando de izquierda a derecha, realizan las siguientes acciones:

- Sube al directorio padre del actual por ejemplo 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 botón **Help** muestra la ayuda que contiene el sistema sobre **kfm**, el uso del programa **kdehelp** se verá más adelante, aunque se puede adelantar que de nuevo es muy similar al de un navegador de **Internet**.
- El semáforo o botón **Stop** permite parar la descarga de ficheros de **Internet**.

### 7.2.2 Crear un nuevo directorio

Una vez en el directorio en el que se quiere crear un nuevo directorio o carpeta se puede 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**. Tras lo que **kfm** abre un cuadro de diálogo preguntando por el nombre del nuevo directorio.

### 7.2.3 Borrar un documento o 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.
- 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**.

Nota: **Linux** no dispone de ningún método que permita recuperar un fichero una vez eliminado. **KDE** permite mover los ficheros a una papelera, esto es a un directorio del que pueden ser eliminados de forma definitiva o recuperados, aunque la funcionalidad de la misma es mucho menor que la de la papelera de **Windows** o **MacOs**.

### 7.2.4 Copiar y mover un documento o directorio

Este proceso es muy sencillo de realizar. 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 nos

permite copiar el elemento (**Copy**), moverlo (**Move**) o crear un enlace (**Link**) con lo cual podemos asignar un nuevo nombre o alias al fichero.

Si en cualquiera de estos casos existe un conflicto, es decir existe un fichero o carpeta con el nombre de los que estamos creando el sistema da la posibilidad de sustituir el fichero o cambiar el nombre del nuevo fichero de forma que no exista el conflicto.

### 7.2.5 Enlaces de KDE

Además de los enlaces que admite el sistema de archivos de **Linux** y que se verá en mayor profundidad más adelante, **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 por ejemplo a un disquete. Cuando se selecciona **New/File System Device** en el menú contextual (botón derecho) o en el menú **File**, se abre un primer cuadro de diálogo en el que se pide el nombre del fichero. Para este ejemplo emplearemos **disquete.kdelnk**. 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 iconos del dispositivo cuando está montado y cuando no lo está. Tras esto cuando se haga 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 **Unmount** 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. El acceso a este tipo de dispositivos por motivos de seguridad es bastante complejo por lo que se expondrá con más detalles más adelante.
- **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 iconos que lo van a representar. Este proceso permite asociar un tipo de fichero a un programa, lo que se verá con mayor detalle en el siguiente capítulo.
- **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 icono. Hay que resaltar que al contrario que en **Windows** las aplicaciones de **Linux** no contienen un icono por lo que se les debe asignar un fichero externo.
- **Internet Address y World Wide Web URL**, similares al tipo FTP salvo que apunta a páginas Web.

### 7.2.6 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 (dependiendo del tipo de fichero), se debe crear un enlace de tipo **Mime Type**, tras lo que el sistema solicitará que se defina un nombre para el nuevo enlace. En el segundo diálogo se pueden definir las propiedades del nuevo tipo, como ejemplo se asociará un fichero de **Microsoft Word** (extensión .doc) con **StarOffice**.

- **Pattern**, en la que podemos especificar las características del tipo de ficheros, por ejemplo \*.zip, \*.htm, \*.gif etc. En el caso del ejemplo se empleará \*.doc.
- **Mime Type**, en el que se define el tipo de fichero por ejemplo text/Word-document
- **Comment**, en el que se puede indicar un texto representativo del tipo de fichero, por ejemplo Documento de **Word**.
- **Default application**, permite la selección de la aplicación que empleará el sistema para abrir el fichero, en el caso del ejemplo **soffice**.

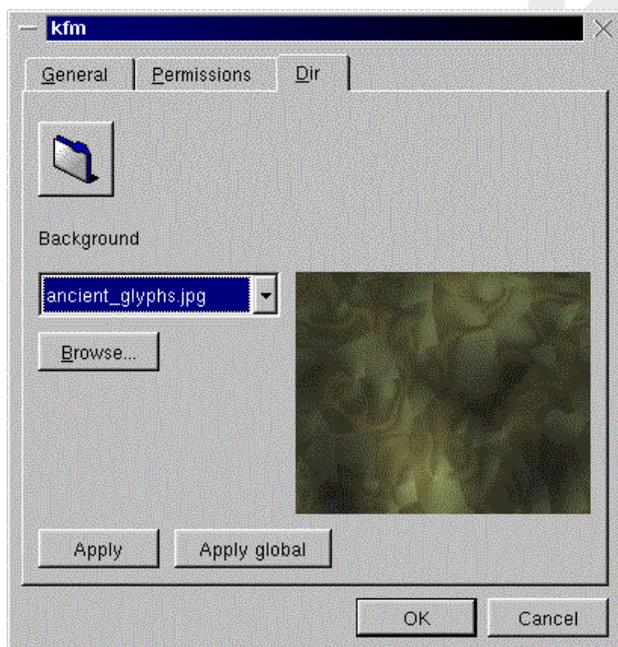


Figura 26

- Además de lo anterior se puede cambiar el icono que mostrará *kfm*.

Además de estas asociaciones manuales *kfm* puede determinar el tipo de un fichero inspeccionando su contenido. Puede así considerar, por ejemplo, una imagen con formato **jpeg** como tal aunque tenga una extensión distinta de **jpg**.

### 7.2.7 Propiedades de un fichero o directorio

*kfm* permite configurar los directorios de forma que se puede seleccionar una imagen de fondo y un icono distinto del que aparece por defecto. El proceso a seguir para realizar estos cambios es muy sencillo. 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 icono. Si por el contrario modificamos 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.

### 7.2.8 Configura *kfm* como navegador de *Internet*

Como se ha comentado al comienzo de este capítulo *kfm* puede actuar como un navegador de *Internet* pudiendo visualizar los ficheros HTML. De hecho 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 *kfm* 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*, para lo que se accede a través del menú *Options/Configure Browser*. En el cuadro de diálogo que se despliega se puede introducir el *proxy* que empleará *kfm* para acceder a *Internet*.

## 7.3 Aplicaciones auxiliares de *KDE*

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

### 7.3.1 *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 del programa *konsole*.

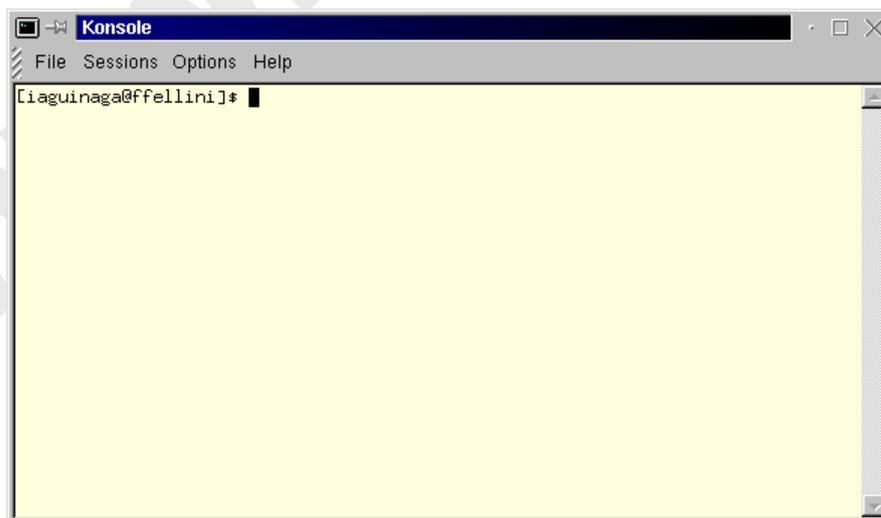


Figura 27

Este programa permite configurar el aspecto de la presentación adaptándola a los gustos del usuario, cambiando el esquema de color, las fuentes el tamaño por defecto de la aplicación, 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 administrados de archivos hasta la consola con lo que se permite copiar el *path* del fichero o cambiar al directorio que contiene un determinado fichero.

### 7.3.2 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*.

*Kedit* admite las opciones típicas de manejo de textos como son copiar un texto (*Edit/Copy*), pegar un texto (*Edit/Paste*) y cortar un texto (*Edit/Cut*), además de otras más sofisticadas como insertar un fichero (*Edit/Insert File*), una fecha (*Edit/Insert Date*), buscar un texto en el documento (*Edit/Find*), reemplazar texto (*Edit/Replace*) o comprobar la ortografía del documento (*Edit/Spellcheck*).

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 (*Options/Font*). Por desgracia la corrección ortográfica sólo está disponible en Inglés y Alemán.

Además de estas opciones *Kedit* es un programa que permite enviar el texto vía mail, editar un fichero a través de un servidor ftp etc.

*Nota:* Para guardar los cambios en la configuración hay que seleccionar el menú *Optios/Save Config*.

### 7.3.3 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*.

Con *kedit* comparte muchas opciones de manipulación de texto con *kedit* (copiar, pegar y cortar, así como buscar y reemplazar texto).

### 7.3.4 kdehelp

Esta aplicación es una de las más interesantes del *KDE* puesto que representa el sistema de ayuda del mismo. Este sistema de ayuda 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.

Como se ha comentado anteriormente la ayuda de **KDE** se basa en **HTML**, por lo que 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 encontramos también en **kfm**, esto es los botones y menús para ir a la página que ha sido visitada anteriormente o con posterioridad, se pueden crear marcadores etc.

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** podemos acceder a la ayuda de ese comando, como ejemplo se puede probar **man: ls**. Dentro de los comandos también se incluyen las funciones de la librería estándar de **C** por lo que **man: sin** o **man: printf** mostrarán la información que contiene el sistema respecto de esas funciones.

### 7.3.5 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:

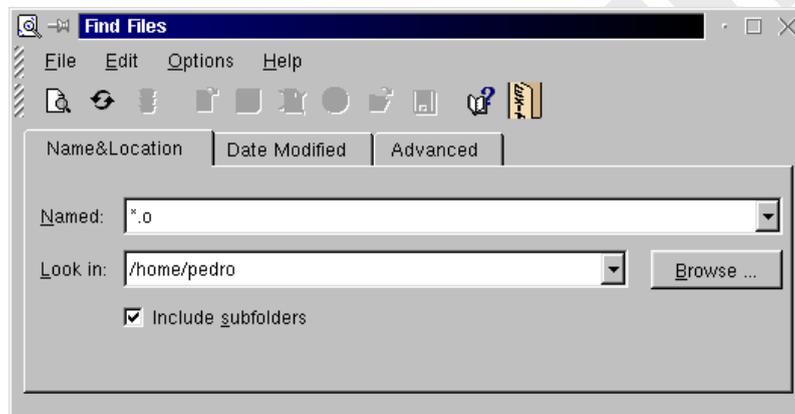


Figura 28

- Por nombre u localización, se conoce el nombre o parte de él y la localización aproximada del fichero.

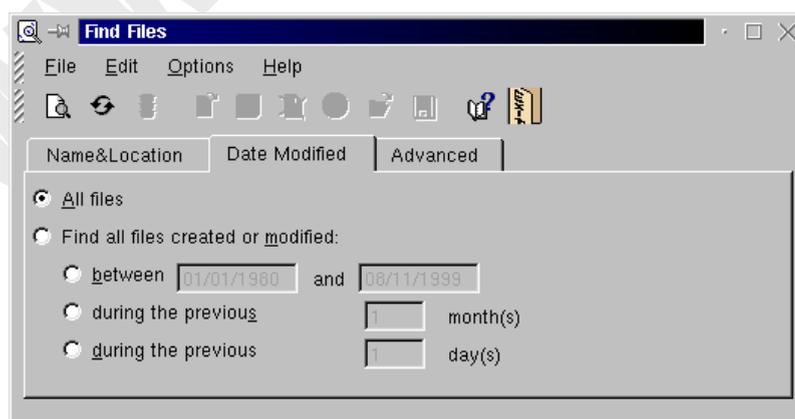


Figura 29

- Se puede centrar aun 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 periodo de tiempo en el que concretar la búsqueda

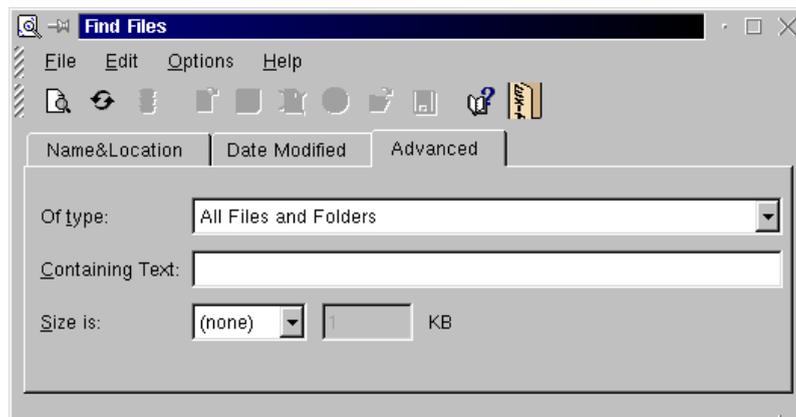


Figura 30

- Por último se puede especificar en **Advanced** que la búsqueda se limite a un determinado tipo de fichero, que el fichero contenga un determinado texto, o que su tamaño sea uno determinado.

Una vez determinados los criterios de selección de los ficheros se puede 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. Tras la búsqueda en la ventana aparecerá una lista con los ficheros coincidentes:

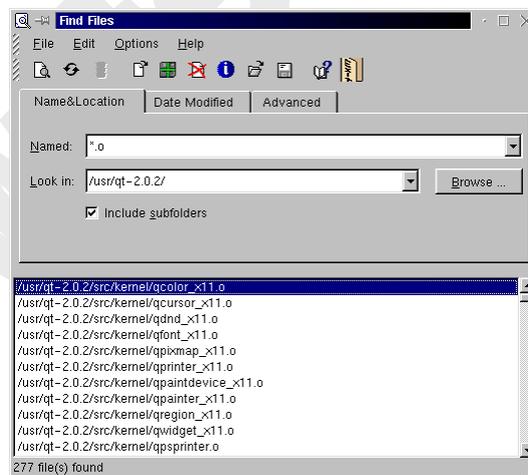


Figura 31

La barra de herramientas de esta aplicación contiene las siguientes funciones empezando por la izquierda:



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11

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

## 7.4 Configuración de *KDE*

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.

### 7.4.1 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 *Utilities/Menu Editor*. Una vez arrancada la aplicación aparecerá una imagen semejante a la siguiente:

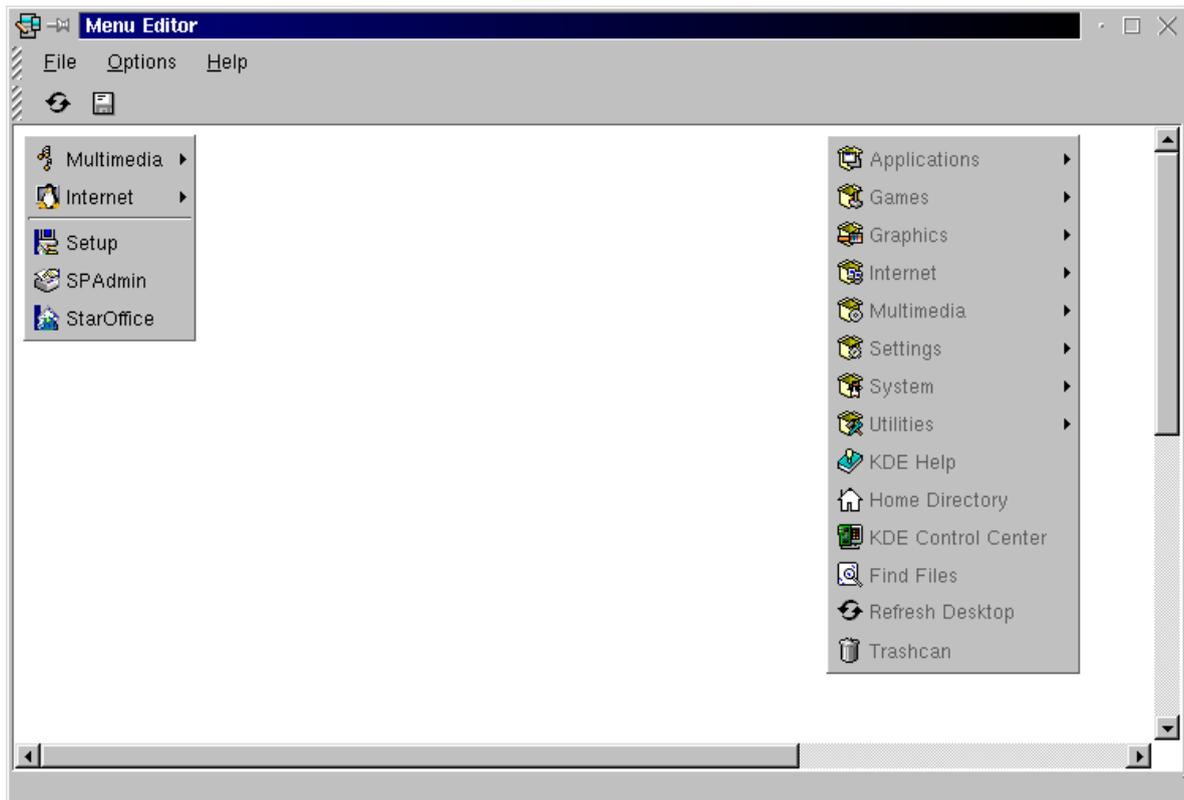


Figura 32

En realidad, y como se aprecia en la figura, el menú de **KDE** está compuesta por dos partes principales. Una de ellas ( a la izquierda) constituye el menú personal del usuario en el que puede añadir o quitar aplicaciones. La segunda de las partes 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.
- **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 el siguiente cuadro de diálogo:

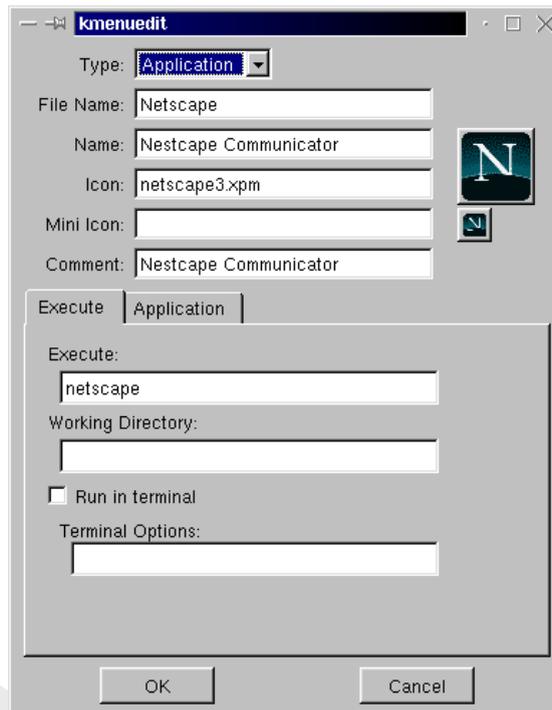


Figura 33

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

- **Type**, tipo del elemento creado puede ser **Separator** ( un separador de distintos elementos), **Submenu** (un submenú), **Application** (una aplicación), **Swallow**, **Link** (un enlace) o **Device** (un dispositivo), 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 (este fichero es un enlace de **KDE** ).
- **Name**, el nombre que aparecerá en el menú una vez creado el mismo

- **Icon**, el nombre de la imagen que aparecerá en el menú. Para seleccionar una imagen se puede clicar en el botón que muestra la imagen lo que abrirá un cuadro de diálogo en el que se puede seleccionar la imagen deseada.
- **Mini Icon**, la imagen que aparecerá cuando sea necesario mostrar un icono 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.

Si lo que se está creando es un enlace a una aplicación en la pestaña **Execute** y la opción **Execute** hay que escribir la línea de comandos necesaria para ejecutar el programa.

#### 7.4.2 KDE Control Center

Esta aplicación es la principal encargada de configurar **KDE** y a ella se puede acceder de muchas formas, tanto desde el icono 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 dividida en dos:

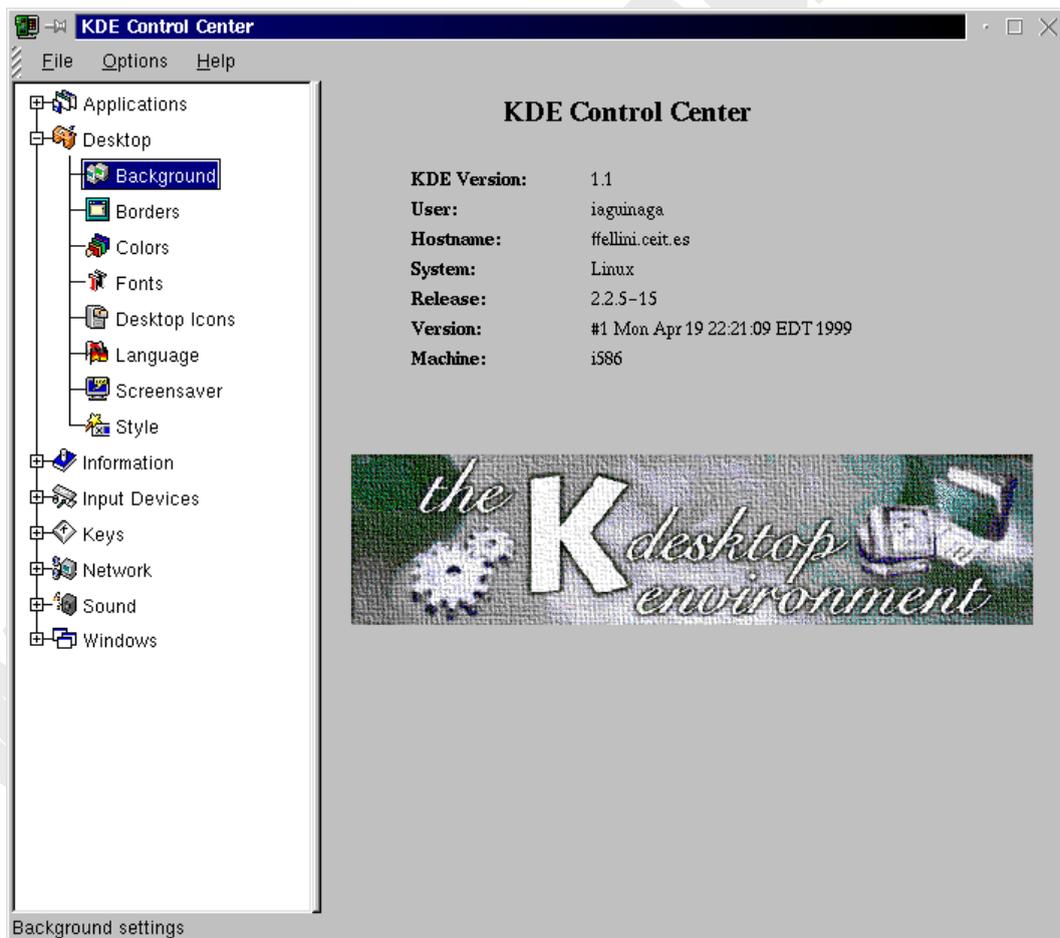


Figura 34

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 abrirán los distintos cuadros de diálogo que permiten configurar *KDE*. Seleccionando la opción *Desktop/Background* aparece a la derecha el cuadro de diálogo que permite cambiar la imagen de fondo de los escritorios virtuales:

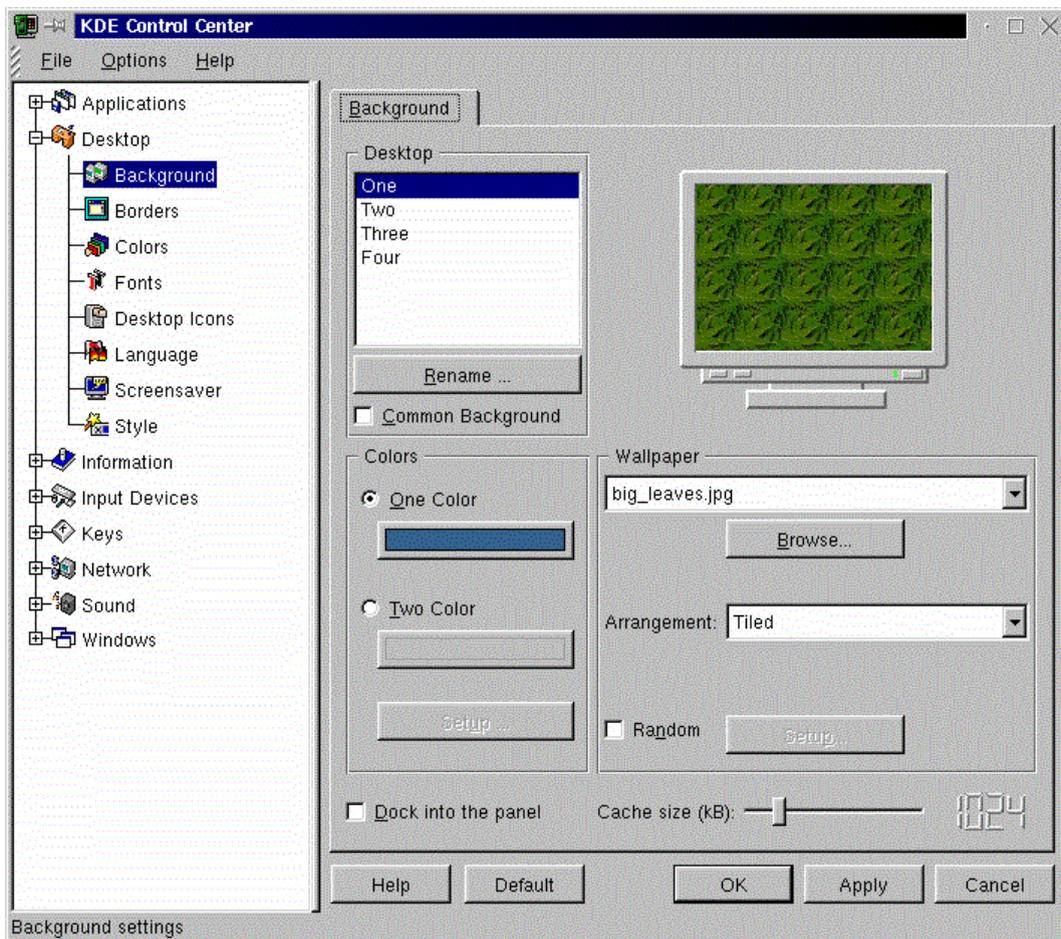


Figura 35

### 7.4.3 Añadir aplicaciones al panel

Otra de las tareas que 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. Seleccionando una cualquiera de ellas esta se añadirá de forma automática al panel.

Para eliminarla o moverla no hay más que hacer clic con el botón derecho del ratón sobre el elemento a modificar y seleccionar la opción pertinente en el menú contextual.

## 7.5 Otras aplicaciones de KDE

Por último se mencionarán algunas de las aplicaciones que están disponibles en *KDE*, y que no se va a explicar su funcionamiento:

- **Korganizer (Applications/Organizer)**, es un programa que permite gestionar la agenda del usuario de una forma sencilla e incluso la sincronización de datos con agendas personales como **PalmPilot**.
- **Icon Editor (Graphics/Icon Editor)**, es un programa de dibujo que permite crear iconos para personalizar los menús y enlaces de **KDE**.
- **Kview (Graphics/Image Viewer)**, es un programa que permite mostrar imágenes de todos los formatos de archivo importantes así como realizar operaciones sencillas con ellas.
- **Paint (Graphics/Paint)**, este es un programa de dibujo básico que permite crear imágenes sencillas.
- **PS Viewer (Graphics/PS Viewer)**, este programa permite visualizar ficheros con imágenes PostScript y documentos de Adobe Acrobat (.pdf).
- **SnapShot (Graphics/Snapshot)**, programa que permite la captura de una ventana y su contenido.
- El menú **Internet** comprende una gran cantidad de programas que se relacionan con **Internet**.
- El menú **Multimedia**, dispone de programas para el visualizado y audición de distintos ficheros multimedia, como pueden ser videos (AVI, Quicktime, MPEG) con **aKtion!**, o escuchar música.
- **Desktop Switching Tool (System/Desktop Switching Tool)** es una 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 (Utilities/Archiver)**, es una aplicación que permite el manejo de ficheros tar y zip de una forma sencilla.
- **Knotes (Utilities/Knotes)**, permite crear notas en el ordenador que luego a las que se puede acceder a través del icono que se añade al panel.

Incluye así mismo una gran cantidad de juegos.

## 8. EL SHELL: COMANDOS BÁSICOS DE LINUX

Hasta este momento se han visto los entornos gráficos existentes para *Linux* más importantes, no obstante cualquier usuario de *Linux* acabará antes o después relacionándose con el sistema empleando el modo texto. Este modo se basa en la ejecución de una serie de comandos, que son interpretados por un programa o *shell*. *Linux* dispone de varios de estos programas pero el más habitual es conocido como *bash* o *Bourne Shell*. Si *Linux* se ha arrancado en modo texto el sistema arranca de forma directa el *shell* y queda a la espera de introducción de nuevos comandos. Si se ha arrancado en modo gráfico se puede acceder al *shell* de dos formas:

- Se puede acceder al *shell* del sistema presionando alguna de las siguientes combinaciones de teclas:
  - `<ctrl>+<alt>+<F1>`
  - `<ctrl>+<alt>+<F2>`
  - `<ctrl>+<alt>+<F3>`
  - `<ctrl>+<alt>+<F4>`
  - `<ctrl>+<alt>+<F5>`
  - `<ctrl>+<alt>+<F6>`

Esto hace que el sistema salga del modo gráfico y acceda a alguna de las seis consolas virtuales de *Linux*, a las cuales también se puede acceder cuando se arranca en modo de texto. Para volver al modo gráfico hay que presionar `<ctrl>+<alt>+<F7>` o `<ctrl>+<alt>+<F8>`.

- La segunda forma es más cómoda y menos radical permitiendo acceder al shell desde el mismo entorno gráfico. Para esto hay que abrir un programa llamado terminal o consola, por ejemplo: *kconsole* (en el entorno *KDE*), *xterm*, *gnome-terminal* (en *GNOME*), etc como se ha visto anteriormente.

### 8.1 Introducción

Existen una serie de nociones básicas que hay que tener en cuenta a la hora de introducir los comandos. En primer lugar citaremos las siguientes:

- Los comandos hay que teclearlos exactamente.
- Las letras mayúsculas y minúsculas se consideran como diferentes.
- En su forma más habitual (los *shells* de Bourne o de Korn), el sistema operativo utiliza un signo de \$ como *prompt* para indicar que está preparado para aceptar comandos, aunque este carácter puede ser fácilmente sustituido por otro u otros elegidos por el usuario. En el caso de que el usuario acceda como administrador este signo se sustituye por #.

- Cuando sea necesario introducir el nombre de un fichero o directorio como argumento a un comando, **Linux**, permite escribir las primeras letras del mismo y realiza un autorrellenado al presionar la tecla del **tabulador**. Si no puede distinguir entre diversos casos rellenará hasta el punto en el que se diferencien. Por ejemplo, supongamos una carpeta con los siguientes directorios:

### **Programas**

#### **Documentos\_proyecto**

#### **Documentos\_privados**

Al escribir **cd Pr<tab> Linux** rellenará el resto del contenido hasta escribir **cd Programas**. Por el contrario al escribir **cd D<tab>** escribirá **cd Documentos\_**

## **8.2 Algunos Comandos Sencillos de LINUX**

Para efectuar el cambio o la introducción de un **password** o **contraseña** se utiliza el comando **passwd**. El proceso a seguir es el siguiente:

### **passwd**

**(current) UNIX password:** (se teclaa la contraseña actual; no aparece en pantalla)

**New UNIX password:** (se teclaa la nueva contraseña; no aparece en pantalla)

**Retype new UNIX password:** (se teclaa de nuevo la nueva contraseña comprobando que se ha teclado bien. Si no coincide no se cambia produce el cambio).

A continuación se describen algunos comandos sencillos de que pueden ser útiles para familiarizarse con los comandos del sistema.

<b>date</b>	Muestra por pantalla el día y la hora.
<b>cal 1949</b>	Muestra el calendario del año 1949.
<b>cal 05 1949</b>	Muestra el calendario de mayo de 1949.
<b>who</b>	Indica qué usuarios tiene el ordenador en ese momento, en qué terminal están y desde qué hora.
<b>whoami</b>	Indica cuál es la terminal y la sesión en la que se está trabajando.
<b>man comando</b>	Todos los manuales de <b>Linux</b> están dentro del propio sistema operativo, y este comando permite acceder a la información correspondiente al comando <b>comando</b> . Por ejemplo con <b>man who</b> aparecerá por pantalla y de forma formateada por páginas, la explicación del comando <b>who</b> . Se puede navegar a través de estas páginas con los cursores del teclado, y presionando <b>q</b> para salir.
<b>clear</b>	Este comando limpia la consola

## 8.3 Directorio Personal

Como se ha visto anteriormente el directorio personal es un directorio con un determinado nombre asignado a un usuario. Los directorios personales habitualmente son subdirectorios de */home* (en algunos casos se utiliza *mnt*, u otro subdirectorio de orden inferior). Generalmente el nombre coincide con el del *nombre de usuario*, aunque puede no ser así, y varios usuarios pueden estar trabajando en el mismo directorio. Cada usuario de *Linux* puede crear una estructura en árbol de subdirectorios y archivos tan compleja como desee bajo su directorio personal pero normalmente nunca fuera de él.

### 8.3.1 Listado del contenido de directorios: comando ls

Una de las acciones más habituales a la hora de trabajar es mostrar el contenido de un directorio, como se ha visto existen herramientas gráficas con este fin, no obstante el *shell* incluye un programa con este mismo fin: *ls*,

<i>ls</i>	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.
<i>ls -a</i>	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 <i>.</i> indica el directorio actual y el doble punto <i>..</i> el directorio padre, que contiene, al actual.
<i>ls -l</i>	Esta 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,...
<i>ls -c</i>	Muestra ordenando por día y hora de creación.
<i>ls -t</i>	Muestra ordenando por día y hora de modificación.
<i>ls -r</i>	Muestra el directorio y lo ordena en orden inverso.
<i>ls subdir</i>	Muestra el contenido del subdirectorio <i>subdir</i> .
<i>ls -l filename</i>	Muestra toda la información sobre el fichero.
<i>ls --color</i>	Muestra el contenido del directorio coloreado.

Las opciones anteriores pueden combinarse. Por ejemplo:

<i>ls -cr</i>	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. Por ejemplo:

- ls \*.gif* Muestra todos los nombres de ficheros que acaben en *.gif*, por ejemplo *dib1.gif, a.gif*, etc.
- ls file?* Muestra todos los ficheros cuyos nombres empiecen por *file* y tengan un nombre de cinco caracteres, por ejemplo: *file1, file2, filea*, etc.

### 8.3.2 Creación de subdirectorios. Comando *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.

### 8.3.3 Borrado de subdirectorios. Comando *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.

### 8.3.4 Cambio de directorio. Comando *cd*

Este comando permite cambiar de directorio a partir del directorio actual de trabajo. Por ejemplo,

*cd /home/Pedro* En este ejemplo pasamos del directorio actual de trabajo al nuevo directorio */home/Pedro*, que será desde ahora nuestro nuevo directorio.

*cd dire* Nos traslada al subdirectorio *dire* (que deberá existir como subdirectorio en el directorio actual).

*cd ..* Retrocedemos un nivel en la jerarquía de directorios. Por ejemplo, si estamos en */home/Pedro* y usamos este comando, pasaremos al escalafón inmediatamente superior de la jerarquía de directorios, en este caso a */home*. **Nota:** al contrario que en *MS-DOS* en *Linux* no existe la forma *cd..* sin espacio entre *cd* y los dos puntos.

*cd* Nos sitúa nuevamente en el directorio personal del usuario.

### 8.3.5 Situación actual. Comando pwd

El comando *pwd* (*print working directory*) visualiza o imprime la ruta del directorio en el que nos encontramos en este momento. Este comando es uno de los pocos que no tiene opciones y se utiliza escribiendo simplemente *pwd*.

### 8.3.6 Acceso a unidades de disco

*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
```

donde *-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. Todos los dispositivos están representados por un fichero del directorio */dev*, por ejemplo en el caso de un disquete será seguramente */dev/fd0*, por último */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*. De todas formas el usuario siempre puede crear un directorio vacío con el nombre que el elija para montar las unidades de disco que desee donde desee.

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 último 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, CD-ROM, etc. hay que editar el fichero */etc/fstab* Por ejemplo para que cualquier usuario pueda acceder a un disquete habrá que indicar la siguiente línea:

```
/dev/fd0 /mnt/floppy vfat user,noauto 0 0
```

También habrá que asegurarse de que el directorio */mnt/floppy* sea accesible por todos los usuarios.

Una vez seguidos los pasos anteriores cualquier usuario podrá "montar" un disquete escribiendo el siguiente comando:

```
mount /mnt/floppy
```

Al igual que antes el usuario deberá ejecutar el comando *umount /mnt/floppy* antes de sacar el disquete.

**Nota:** 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.

### 8.3.7 Copia de ficheros. Comando **cp**

Este comando tiene la siguiente forma,

```
cp file1 file2
```

y hace una copia de *file1* y le llama *file2*. Si *file2* no existía, lo crea con los mismos atributos de *file1*. Si *file2* existía antes, 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 el la ruta al mismo si alguno de ellos no se encuentra en el directorio actual. Otra posibilidad es:

```
cp file1 file2 namedir
```

que hace copias de *file1* y *file2* en el directorio *namedir*.

### 8.3.8 Traslado y cambio de nombre de ficheros. Comando **mv**

Este comando tiene una forma similar al anterior ,

```
mv file1 file2
```

El comando *mv* realiza la misma función que el anterior (*cp*) pero además destruye el fichero original. En definitiva traslada el contenido de *file1* a *file2*; a efectos del usuario lo que ha hecho es cambiar el nombre a *file1*, llamándole *file2*. De igual forma,

```
mv file1 file2 namedir
```

traslada uno o más ficheros (*file1, file2,...*) al directorio *namedir* conservándoles el nombre. El comando,

```
mv namedir1 namedir2
```

cambia el nombre del subdirectorio *namedir1* por *namedir2*.

Hay que recalcar que el comando *mv* sirve así mismo para cambiar el nombre de los ficheros.

### 8.3.9 Enlaces a ficheros. Comando **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.

***ln panacea subdir/panacea***

Después de este comando el fichero ***panacea*** tendrá el mismo nombre, pero a efectos del usuario estará colocado en dos sitios distintos: en el directorio actual y en el subdirectorio ***subdir***.

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

### 8.3.10 Borrado de ficheros. Comando ***rm***

Este comando tiene las formas siguientes,

***rm file1 file2***

Este comando elimina uno o más ficheros de un directorio en el cual tengamos 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. Por ejemplo, si se teclea,

***rm -i superfluo***

aparecerá en pantalla el aviso siguiente:

***remove superfluo?***

y habrá que contestar ***y*** (yes) o ***n*** (not). En este comando se pueden utilizar los caracteres de sustitución (***\**** y ***?***), como por ejemplo,

***rm fich\****

que borraría todos los ficheros del directorio actual que comiencen por ***fich***. El comando

***rm \****

borraría todos los ficheros del directorio actual, mientras que

***rm -i \****

realiza una labor análoga, pero con previa confirmación.

### 8.3.11 Características de un fichero. Comando **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.

### 8.3.12 Cambio de modo de los ficheros comandos **chmod**, **chown** y **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*** Nombres de los ficheros cuyos modos de acceso se quieren cambiar.

Por ejemplo, para quitar el permiso de lectura a los usuarios de un fichero el comando es:

***chmod a -r fichero.txt***

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 (“**change owner**”) 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 del comando **chown** 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...***

Los grupos de usuarios están almacenados en el fichero ***/etc/group***.

#### 8.4 Espacio ocupado en el disco Comandos ***du*** y ***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***, éste comando da 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.

#### 8.5 Visualización sin formato de un fichero. Comando ***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***). Para terminar se emplea ***<ctrl>d***

#### 8.6 Comando ***head***

***head -7 filename***

escribe las 7 primeras líneas del fichero ***filename***

#### 8.7 Visualización de ficheros con formato. Comando ***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 en relación con el comando ***lp*** de salida por impresora. Las formas más importantes que admite son las siguientes:

<b><i>pr file</i></b>	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).
<b><i>pr -ln file</i></b>	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)
<b><i>pr -p file</i></b>	Hace una pausa para presentar la página, hasta que se pulsa <code>&lt;return&gt;</code> para continuar
<b><i>pr -t file</i></b>	Suprime las 5 líneas del encabezamiento y las del final de página.
<b><i>pr -wn file</i></b>	Ajusta la anchura de la línea a n posiciones.
<b><i>pr -d file</i></b>	Lista el fichero con espaciado doble.
<b><i>pr -h `caracteres` file</i></b>	el argumento o cadena de caracteres <code>`caracteres`</code> se convertirán en la cabecera del listado.
<b><i>pr +n file</i></b>	Imprime el fichero a partir de la página n.

Además de los ejemplos anteriores, se pueden combinar varias opciones en un mismo comando, como por ejemplo en: ***pr -dt file*** la salida de este comando es por la consola, pero puede redireccionarse a otro fichero, por ejemplo, si ejecutamos el comando: ***pr file1 > file2*** se crea un fichero nuevo llamado ***file2*** que es idéntico a ***file1***, pero con formato por páginas y columnas.

## 8.8 Visualización de ficheros pantalla a pantalla. Comandos **more** y **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.

## 8.9 Búsqueda en ficheros. Comandos **grep**, **fgrep** y **egrep**

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
```

siendo 'conjuntocaracteres' la secuencia de caracteres a buscar, y *file1*, *file2*, y *file31* los ficheros donde se debe buscar. Veamos un nuevo ejemplo:

```
grep 'TRIANGULARIZACION MATRIZ' matrix.f scaling.f
```

Este comando buscará **TRIANGULARIZACION MATRIZ** entre las líneas de los ficheros *matrix.f* y *scaling.f*. 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:

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

A continuación se muestra una serie de ejemplos.

- **grep '^d' text** líneas que comienzan por d.
- **grep '[^d]' text** líneas que no comienzan por d.
- **grep -v '^C' file1 > file2** quita las líneas de file1 que comienzan por C y lo copia en file2.

## 8.10 Comandos tar y gzip

Tanto el comando **tar** como **gzip** 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 etc. 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 que agrupa varios ficheros en uno, **gzip** comprime un único fichero con lo que la información se mantiene pero se reduce el tamaño del mismo. El uso de **gzip** es muy sencillo

### **gzip fichero**

con lo que se comprime fichero (que es borrado) y se crea un fichero con nombre **fichero.gz**. Si lo que se desea es descomprimir un fichero se emplea entonces:

### **gzip -d fichero.gz**

recuperando el fichero inicial. Como se ha comentado al principio es típico emplear **tar** y **gzip** de forma consecutiva, para obtener ficheros con extensión **tar.gz** o **tgz** que contienen varios ficheros de forma comprimida (similar a un fichero **zip**). El comando **tar** incluye la opción **z** para estos ficheros de forma que para extraer los ficheros que contiene:

### **tar -zxf fichero.tar.gz**

## **8.11 Comandos de impresión.**

### **8.11.1 Comando 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.

## 9. REDIRECCIONES Y TUBERÍAS

### 9.1 Redirecciones

Los comandos de *Linux* tienen una entrada estándar (número 0) y dos salidas estándar (número 1 para la salida normal del comando, y número 2 para la salida de los mensajes de error que se puedan producir en su ejecución). Por defecto tanto la entrada como las salidas estándar de los comandos son la propia terminal, a no ser que por la propia naturaleza del comando se den en él los nombres de algunos ficheros que hagan el papel de entrada y de salida. Por ejemplo, en el comando

```
cp file1 file2
```

*file1* es la entrada y *file2* es la salida; aquí no intervienen las entradas y salidas estándar. Sin embargo, cuando utilizamos por ejemplo el comando *ls* (listado de directorio), la salida de este comando se dirige hacia la terminal. Si queremos que la salida de este comando se dirija a un fichero llamado *file*, podríamos escribir,

```
ls >file
```

el (>) es uno de los llamados operadores de redirección y dirige la salida estándar hacia el fichero indicado a continuación; si este fichero no existe, se crea en ese momento. Otros operadores de redirección son el operador (<) que redirige la entrada estándar desde un determinado fichero, y el operador (>>) que redirige la salida estándar hacia otro fichero, pero añadiendo dicha salida al final de ese fichero, sin sobrescribir el contenido original. Por ejemplo, si cada vez que entramos en el sistema ejecutamos el comando,

```
date >>archivo
```

tendremos un fichero llamado *archivo* que contiene información sobre todas las veces que hemos entrado en el sistema. Otro ejemplo, para añadir al fichero *file2* al final de *file1* y al conjunto llamarle *file3*, sería

```
cat file1 file2 >file3
```

o, si quisiéramos que el fichero resultante fuera el mismo *file1*,

```
cat file2 >>file1
```

Un ejemplo en redirección a la entrada podría ser el siguiente,

```
mail juan <carta
```

que envía al usuario *juan* el contenido del fichero *carta*.

### 9.2 Tuberías

Siguiendo con los ejemplos anteriores, si quisiéramos enviar a *juan* una lista de nuestros ficheros podríamos utilizar los comandos,

```
ls >fichero
```

```
mail juan <fichero
```

```
rm fichero
```

Es decir que hemos conectado la salida estándar de *ls* con la entrada estándar de *mail*, a través de un fichero transitorio *filelist*. *Linux* permite hacer esta operación directamente, sin pasar por el fichero de almacenamiento transitorio: esto se hace mediante el concepto de tubería (pipe), que consiste en empalmar la salida estándar de un comando con la entrada estándar de otro. Para el ejemplo anterior esto se hace en la forma,

```
ls | mail juan
```

Con el operador de tubería (*|*) se pueden empalmar tantos comandos como se desee.

### 9.3 Bifurcación o T (comando tee)

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 podría emplearse de la siguiente forma:

```
ls | tee file
```

la salida de *ls* se bifurca hacia la terminal y hacia *file*.

Si quisiéramos que la salida de este comando se añadiera al final de *file*, deberíamos utilizar la opción *-a*,

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

### 9.4 Redirección de la salida de errores

Los mensajes de error se dirigen a la salida número 2, que normalmente es también la terminal. A veces, por ejemplo cuando se quiere ejecutar un comando en background (ejecutar un comando en background es lanzar su ejecución y recuperar el control de la terminal sin esperar a que termine, lo cual se hace añadiendo el carácter *&* al final del comando), interesa evitar que los mensajes de error aparezcan en la pantalla, pues en ella habremos empezado a hacer otra cosa.

Supongamos por ejemplo que queremos compilar y montar en background un conjunto de ficheros, dirigiendo los listados a un fichero llamado *listados*, y los mensajes de error a un fichero llamado *errores*. Lo haríamos en la forma,

```
gcc prueba.c 2>errores
```

con lo cual la salida 2 (errores) se redirige hacia el fichero *errores*. Para redirigir la salida estándar de errores al mismo fichero que la salida estándar se emplea un comando como:

```
program <datos.d >resultados.r 2>&1
```

## 10. EJECUCIÓN DE PROGRAMAS

### 10.1 Ejecución en el fondo & , kill, nice y nohup

Para ejecutar un programa en el fondo, es decir, recuperando inmediatamente el control del terminal, basta añadir el carácter **&** al final del comando de ejecución:

```
program <datos.d >resultados.r &
```

inmediatamente aparecerá en el terminal, debajo de esta línea, un número que es el **número de proceso** de la ejecución de este programa. Para detener definitivamente dicha ejecución (no se puede detener temporalmente) se puede utilizar el comando **kill**:

```
kill númerodeproceso
```

La ejecución de un programa en el fondo no impide que aparezcan en la pantalla los mensajes de error que se produzcan (a no ser que se haya redirigido la salida de errores), y que el programa se pare cuando se salga del sistema. Para que el programa continúe ejecutándose aún cuando nosotros hayamos terminado la sesión, hay que utilizar el comando **nohup**:

```
nohup program
```

Si no se utilizan redirecciones todas las salidas del programa se dirigen a un fichero llamado **nohup.out**. Cuando se utiliza **nohup** el ordenador entiende que el usuario no tiene prisa y automáticamente disminuye la prioridad de la ejecución. Existe un comando, llamado **nice**, que permite realizar ejecuciones con baja prioridad, es decir se le indica al ordenador que puede ejecutar de forma más lenta esta aplicación si existen otras que sean más urgentes. Se utiliza en las formas,

```
nice program &
```

```
nice nohup program &
```

Para darle al programa la prioridad mínima habría que utilizar el comando,

```
nice -19 program &
```

donde el -19 indica la mínima prioridad.

### 10.2 Comando time

El comando **time**, precediendo a cualquier otro comando, suministra información acerca del tiempo total empleado en la ejecución, del tiempo de CPU utilizado por el programa del usuario, y del tiempo de CPU consumido en utilizar recursos del sistema. Por ejemplo para saber el tiempo utilizado en la compilación y montaje del programa **prueba.c** utilizaríamos el comando,

```
time gcc prueba.c
```

### 10.3 Comando top

**Linux** incluye una aplicación llamada **top** cuya finalidad es manipular la ejecución de programas de una forma interactiva. Esta aplicación muestra una lista de los procesos que se están ejecutando. Los principales comandos de top son: u que muestra los procesos que pertenecen a un determinado usuario, k equivalente al comando kill para matar un proceso y h que muestra la ayuda del programa.

## 11. PROGRAMAS DE COMANDOS

El sistema operativo *Linux*, al igual que otros sistemas operativos, permite realizar programas de comandos, esto es, programas constituidos por distintos comandos que podrían teclearse interactivamente uno por uno en una terminal, pero que es muchas veces más cómodo agruparlos en un fichero, y ejecutarlos con una sola instrucción posteriormente.

Los comandos de *Linux* pueden ser *externos* - que implican la creación de un nuevo proceso, cuyo código está en */bin* o */usr/bin*- e *internos* - cuyo código está incluido en el del intérprete shell que los ejecuta.

Una cierta primera forma de agrupar comandos la ofrece *Linux* por medio del carácter *;*. Por ejemplo, tecleando el comando,

*date; ls; who*

el ordenador ejecutará sucesivamente los comandos *date*, *ls* y *who*. También podría crearse con un editor de textos un fichero llamado *comandos* que contuviera las líneas siguientes:

*date*

*ls*

*who*

Para ejecutar este fichero de comandos puede teclearse,

*sh comandos*

o bien convertir el fichero *comandos* en directamente ejecutable por medio del comando *chmod* en la forma,

*chmod a+x comandos*

de modo que el programa de comandos *comandos* puede ejecutarse simplemente tecleando su nombre,

*comandos*

Los comandos *sh comandos* y *comandos* no son enteramente equivalentes. Así, el primero de ellos exige que el fichero *comandos* esté en el directorio de trabajo, mientras que el segundo sólo exige que el fichero *comandos* esté en uno de los directorios de búsqueda de comandos especificados en la variable *PATH*.

Cuando se ejecuta un fichero de comandos *Linux* abre lo que se llama un nuevo shell, es decir un nuevo entorno para la ejecución de los comandos. Para que las variables del caparazón original conserven su valor en el nuevo caparazón es necesario prepararlas con la sentencia *export* antes de abrir el nuevo shell. Por ejemplo, como consecuencia de lo que se acaba de decir, si en el interior de un fichero de comandos se cambia de directorio con el comando *cd*, al acabar la ejecución de dicho fichero volveremos automáticamente al directorio inicial.

## 11.1 Introducción de comentarios

Para introducir líneas de comentarios en un programa de comandos basta comenzar dichas líneas con el carácter `#`. Hay que tomar la precaución de que este carácter no sea el primer carácter del fichero de comandos, porque entonces el ordenador interpreta que el programa está escrito en *C-shell* (una variante especial de UNIX desarrollada en la Universidad de Berkeley) y el resultado es imprevisible. Puede ser buena práctica comenzar todos los ficheros de comandos con una línea en blanco.

## 11.2 Variables del shell

UNIX permite definir variables en un fichero de comandos en la forma,

```
USER=/mnt/mecan/juanto
```

```
TERM=hp2392
```

```
...
```

Es una práctica habitual el utilizar nombres con letras mayúsculas para las variables del caparazón. Para recuperar el valor de una variable hay que precederla con el carácter `.` Por ejemplo, utilizando en otra parte del programa *TERM*, en dicho lugar se sustituiría *TERM* por su valor, esto es, hp2392.

El shell del *Linux* tiene definidas para cada usuario unas variables estándar. Para averiguar cuáles son basta teclear el comando siguiente,

```
set
```

Para definir otras variables propias de cada usuario puede utilizarse el fichero *.profile*, que es un fichero de comandos propio de cada usuario que se ejecuta automáticamente al hacer el login.

Para definir variables que contengan espacios en blanco deben encerrarse entre caracteres (`'`) o (`"`), como por ejemplo,

```
FECHA="31 de Diciembre de 1986"
```

más adelante se verá la diferencia entre el carácter (`'`) y el carácter (`"`).

## 11.3 Comando echo

El comando *echo* imprime un determinado texto en la terminal. Un ejemplo de utilización de dicho comando puede ser el siguiente:

```
echo Me gusta el sistema operativo UNIX
```

El comando *echo* es de gran utilidad en los ficheros de comandos. Cuando el texto que se desea escribir en la terminal contiene alguno de los caracteres especiales de UNIX (`* ? [ ] > >> < & ; \ '` ) hay que tomar precauciones especiales desconectando su significado. Una forma de hacerlo es precediendo dicho carácter con la barra invertida (`\`). Así, para escribir mediante el comando *echo* tres asteriscos, utilizaríamos

```
echo |*|*|*
```

si no utilizáramos la barra invertida, el asterisco se interpretaría como un carácter de sustitución y se imprimiría el nombre de todos los ficheros del directorio.

Otra forma de anular el significado de los caracteres especiales es encerrando el texto a escribir mediante comillas (") o entre apóstrofes normales ('). Los apóstrofes (') anulan el significado de todos los caracteres comprendidos entre ellos. Así pues, el triple asterisco lo podríamos escribir con el comando,

```
echo '***'
```

Las comillas (") son menos restrictivas, y anulan el significado de todos los caracteres excepto los tres siguientes: ( ` \). Esto es muy importante porque si VAR es el nombre de una variable, y VAR aparece en un comando echo entre apóstrofes se escribe VAR, mientras que si aparece entre comillas se escribe el valor de la variable, al cumplir el carácter su cometido.

El carácter (\) tiene otros significados, además del ya visto de anular el significado especial de otros caracteres. Así, sirve como indicador de que un comando continúa en la línea siguiente. Cuando se utiliza en la definición interactiva de un comando, en la línea siguiente aparece el prompt secundario (>), que indica que se debe seguir tecleando el comando. Cuando en un comando echo aparecen los caracteres (\c) y (\n) quiere decir, respectivamente, que no se cambie de línea y que se salte de línea, al escribir por la pantalla.

El carácter apóstrofo inverso o acento grave (`) tiene también un significado especial. Cuando en un comando echo aparece el nombre de otro comando encerrado entre apóstrofes inversos (por ejemplo, `date`, `who`, `ls`, ...), el nombre de dicho comando se sustituye por el resultado que genera al ejecutarse interactivamente. Un ejemplo podría ser el siguiente:

```
echo "Los usuarios del sistema son \n\n `who`"
```

El lector puede hacer la prueba y observar el resultado correspondiente.

#### 11.4 Parámetros de los ficheros de comandos

A los ficheros de comandos pueden pasárseles como parámetros un conjunto de una o más variables. Dentro del fichero de comandos estas variables o parámetros se conocen con los nombres **0**, **1**, **2**, ..., **9**. La variable **0** representa el propio nombre del fichero de comandos, y **1**, **2**, ..., **9** son los nombres de los parámetros propiamente dichos

Vamos a comenzar viendo un ejemplo muy sencillo de programa de comandos al que se le pasa sólo una variable o parámetro. El comando de borrar de **Linux rm** no confirma la operación de borrado si no se le pone la opción (**-i**). Esto es peligroso porque uno fácilmente puede olvidarse de teclear dicha opción y borrar lo que no quería borrar. Vamos a crear un fichero de comandos llamado **del** que incluya dicha opción. Dicho fichero podría estar formado por,

```
echo "Quiere borrar el fichero 1?"
```

```
rm -i 1
```

Después de darle a este fichero el correspondiente permiso de ejecución con el comando **chmod**, podríamos borrar con confirmación el fichero **file** tecleando,

**del file**

Dentro del fichero de comandos, **0** valdría **del** y **1** valdría **file**.

Un programa de comandos más complicado y que utiliza dos parámetros podría ser el contenido en el fichero **cambio**, que intercambia el nombre de dos ficheros:

**mv 1 ficheropufo**

**mv 2 1**

**mv ficheropufo 2**

Este fichero se ejecutaría en la forma,

**cambio file1 file2**

En este ejemplo **0** es **cambio**, **1** es **file1** y **2** es **file2**. En realidad a un fichero de comandos se le pueden pasar todos los argumentos que se deseen, aunque sólo hay nombre específico para los nueve primeros (más el propio nombre del comando). El número de argumentos que se le pasa está contenido en la variable **#**. La variable **\*** contiene el conjunto de todos los parámetros. Un nuevo ejemplo puede aclarar algo más este punto.

Si el programa **del** que hemos hecho previamente lo hubiéramos utilizado en la forma:

**del \*.f**

teóricamente debería de borrar, con confirmación, todos los ficheros **Fortran** del directorio. En la práctica no es así, porque **(\*.f)** no representa un único argumento sino muchos argumentos (todos los ficheros **Fortran** del directorio). Como resultado sólo se borra el primer fichero **Fortran**. Para borrar todos e indicarnos además cuántos ficheros hay, el fichero **del** podría estar compuesto por los siguientes comandos:

**echo "Hay # programas Fortran \n"**

**rm -i \***

El comando **shift** hace posible utilizar y distinguir parámetros que están más a la derecha del noveno lugar en la llamada al programa de comandos. En efecto, cuando se llama al comando **shift**, **2** se convierte en **1**, **3** en **2**, etc, y lo que hubiera sido **10** en **9**, con lo cual ya se puede referenciar. El comando **shift** deja inalterado **0** y puede utilizarse tantas veces como se desee.

## 11.5 Otras posibilidades de los ficheros de comandos

Los ficheros de comandos tienen muchas más posibilidades que las que se han apuntado en esta Introducción: pueden leer variables, preguntar por la existencia de un fichero y por si es ejecutable o no, y

admiten construcciones lógicas del tipo **IF**, **DO**, **DO WHILE**, etc. Para utilizar estas posibilidades acudir al manual correspondiente.

[www.technun.es](http://www.technun.es)

## 12. ANEJOS

### 12.1 Compilado de programas en Linux

**Linux** como cualquier sistema **Unix** que se precie incluye un compilador de **C** y **C++**. Esto no implica que se esté limitado a estos dos lenguajes de programación. Por el contrario existen una gran cantidad de compiladores gratuitos para los lenguajes más importantes.

El compilador de **C/C++** de **Linux** es un compilador de línea de comandos, esto es, no dispone de ninguna interfaz gráfica que facilite la programación y compilación del código. Existen eso sí editores de texto capaces de mostrar la sintaxis del código resaltada como **kwrite**, aunque la compilación hay que realizarla manualmente a través de una consola o terminal.

### 12.2 Compilación y linkado

El primer paso para crear un programa, por ejemplo en C++, es crear el fichero de código fuente, y guardarlo en un fichero de texto por ejemplo **e1.cpp**. Tras esto hay que compilar el programa. Para esto se empleará el comando **g++**, de la siguiente forma:

```
g++ e1.cpp
```

Con lo que conseguiremos que se compile el programa. Con este comando hemos conseguido que se cree un programa llamado **a.out** en el directorio de trabajo. Para ejecutarlo emplearemos

```
./a.out
```

Normalmente no desearemos que el programa se llame **a.out** sino que tenga un nombre más significativo, como **ejemplo\_1**, para conseguirlo emplearemos:

```
g++ -o ejemplo_1 e1.cpp
```

Si queremos ejecutar el programa emplearemos **./ejemplo\_1**. Si el programa escrito realiza emplea alguna función que no se encuentre en la librería estándar hay que incluirla en la orden de compilación, por ejemplo:

```
g++ -o ejemplo_2 e2.cpp -lname
```

Donde **name** es el nombre de la librería. Por defecto estas se guardan en un fichero cuyo nombre es **libname.so**, cuya localización suele ser **/usr/lib**.

Si el programa no está escrito en C++ sino en C las opciones de compilación son las mismas salvo que en lugar del programa **g++** se empleará el comando **gcc**.

```
gcc -o ejemplo_3 e3.c
```

### 12.3 Comando make

Este comando sirve para organizar la compilación y el enlazado de programas complicados que dependen de muchos módulos y librerías diferentes. Cuando se ejecuta este comando, se construye un nuevo ejecutable volviendo a compilar sólo aquellos ficheros fuente que son más recientes que el los ficheros compilados correspondientes, teniendo en cuenta para ello las fechas de última modificación de cada fichero.

Este comando se basa en un fichero ASCII (llamado por defecto *makefile*) que contiene una relación de dependencias entre los distintos módulos, así como las acciones que hay que realizar para poner a punto cada módulo, es decir para pasar de un fuente a un objeto, por ejemplo. Este comando tiene la siguiente forma general:

***make [-f makefilename] [-arg\_opt] [exe\_name]***

El fichero *makefile* (con éste o con otro nombre invocado por medio de la opción *-f*) contiene cuatro tipos de líneas diferentes:

- Líneas de *comentario*, que comienzan por el carácter (#). Si en una línea cualquiera aparece el carácter (#), se ignora todo lo que aparece a continuación de dicho carácter en dicha línea.
- Líneas de *definición de macros*. Tienen la forma general,

***IDENTIFICADOR = cadena\_de\_caracteres***

Si en alguna otra línea aparece (IDENTIFICADOR), dicha ocurrencia se sustituye por *cadena\_de\_caracteres*. No es necesario que el nombre del identificador esté escrito con mayúsculas, pero es una costumbre bastante extendida el hacerlo así. Mediante el uso de macros se pueden representar brevemente pathnames o listas de nombres de ficheros largos. Si el identificador tiene una sola letra, no hace falta poner los paréntesis. El comando *make* tiene una serie de macros definidas por defecto que se pueden listar con el comando *make -p*.

Líneas describiendo las *relaciones de dependencia*. Tienen la forma,

***file.o fila.o ... : file1.cpp file2.cpp ...***

La lista de ficheros que están a la izquierda del carácter (:) dependen de los ficheros que están a la derecha. En estas líneas se realiza la sustitución habitual de los caracteres (?\*[])

Líneas de *comandos shell*, comenzando siempre por un tabulador. Estas líneas representan las acciones que hay que realizar para actualizar los ficheros dependientes, según las relaciones de dependencia descritas en la línea anterior. En una misma línea de comandos puede haber varios comandos separados por un carácter (;), y de este modo se ejecutan en un mismo proceso; si hay varias líneas de comandos, cada línea se ejecuta en un proceso diferente. Estos comandos shell (compilaciones, etc.) se ejecutan o no según las fechas de los ficheros correspondientes. Hay también una lista de dependencias implícitas y de macrodefiniciones standard que se pueden obtener con el comando *make -p* (la salida de este comando puede tener varios cientos de líneas). A continuación se presentan algunos ejemplos de ficheros *makefile*.

A continuación se muestran una serie de ejemplos

```
# Esto es un ejemplo de fichero makefile
# Definiciones de macros
ma = ma27ad.o ma27bd.o ma27cd.o

TEST: test.o (ma)
      xlf -o TEST -O (ma)

test.o: test.f
      xlf -c -O test.f
ma27ad.o: ma27ad.f
      xlf -c -O ma27ad.f
ma27bd.o: ma27bd.f
      xlf -c -O ma27bd.f
ma27cd.o: ma27cd.f
      xlf -c -O ma27cd.f
```

Hay informaciones que no es necesario dar por que se suponen por defecto. Por ejemplo, si el fichero objeto (.o) y el fichero C (.c) o C++ (.cpp) tienen el mismo sufijo, no hace falta definir esas relaciones de dependencia, que por otra parte son triviales.

## 12.4 Búsqueda avanzada en ficheros. Expresiones regulares

A veces se desea encontrar las líneas de un fichero que contienen una palabras o palabras determinadas. Cuando el texto que se desea encontrar es único, lo que hay que hacer es ponerlo tal cual en la sección del comando que define la búsqueda, por ejemplo

```
grep "PATATAS" Lista_de_la_compra.txt
```

Sin embargo, en otras ocasiones el texto que se desea buscar no es único, es decir, no está unívocamente determinado como en el ejemplo anterior, sino que debe cumplir unas ciertas condiciones, como la de estar escrito con mayúsculas, comenzar por determinado carácter, estar a principio o final de línea, etc. Este problema se puede resolver en muchos comandos de *Linux* por medio de las expresiones regulares que se van a presentar a continuación.

Las *expresiones\_regulares* son una forma de describir patrones para la búsqueda de unas determinadas líneas dentro de uno o más ficheros ASCII. Se trata pues de encontrar las líneas cuyo contenido cumple ciertas condiciones, que se definen en la *expresión\_regular*.

### 12.4.1 Caracteres especiales

En una *expresión\_regular* se pueden utilizar algunos caracteres que tienen un significado especial. Son los siguientes:

- [        comienzo de la definición de un conjunto de caracteres
- .        un carácter cualquiera, excepto el <eol>

\* un conjunto de caracteres cualesquiera, excepto el primer carácter de una expresión o inmediatamente después de la secuencia \`(`

`)` terminación de la definición de un conjunto de caracteres

- sirve para definir el conjunto de caracteres que van del que le precede al que le sigue. Si va detrás del `[` o delante del `]`, no es especial

`^` comienzo de línea, si está al comienzo de la expresión

`^` conjunto complementario (el que no cumple la condición), si está justo después del `[` que abre la definición de un conjunto

`$` fin de línea, cuando está al final de una expresión

`\` quita el significado especial a un carácter, excepto en la definición de un conjunto de caracteres

### 12.4.2 Expresiones regulares de un solo carácter

Se trata de buscar palabras o conjuntos de un solo carácter, que cumple ciertas condiciones. A continuación se presentan algunos ejemplos:

`\*` representa el carácter `*`

`.` cualquier carácter, excepto el `<eol>`

`[a-f]` un carácter cualquiera entre la `a` y la `f`

`[A-Z]` cualquier letra mayúscula

`[^a-d]` cualquier carácter que no sea una letra entre la `a` y la `d`

`[:clase:]` donde *clase* puede ser: *digit* (cifra del 0 al 9), *xdigit* (cifra hexadecimal), *alpha* (letra cualquiera), *upper* (letra mayúscula), *lower* (letra minúscula), *alnum* (letra o dígito cualquiera), *space* (un espacio en blanco), *cntrl* (carácter de control), *punct* (un carácter de puntuación) y *print* (carácter imprimible).

### 12.4.3 Expresiones regulares generales

Se pueden formar de acuerdo con las siguientes reglas:

- una expresión regular de un sólo carácter

`[a-z]` cualquier letra minúscula

- una expresión regular de un sólo carácter, seguida del carácter `*`, representando entonces todas las palabras de longitud positiva o nula que se pueden construir con los caracteres aceptados por la una expresiones regulares de un sólo carácter

[a-z]\* cualquier palabra escrita con minúsculas

- concatenando (poniendo una a continuación de la otra) dos expresiones regulares construidas previamente

[a-z][A-Z] **cualquier palabra de dos letras, de las cuales la primera es minúscula y la segunda mayúscula**

- Una expresión\_regular definida en la forma `\(expresiones regulares\)` representa la propia expresiones regulares (es decir, definida ella sola), pero define una forma de referirse luego a esa expresiones regulares. En efecto las expresiones regulares definidas de esta forma quedan afectadas por un número del 1 al 9, y es posible luego hacer referencia a una expresiones regulares por medio del número que le corresponde, en la forma `\número`. Si `\número` va seguido de un `*`, esa subexpresión puede repetirse un número cualquiera de veces.

`^\(.*\)1\1` Al comienzo de la línea, un campo formado por un carácter cualquiera que se repite las veces que sea, volviendo a aparecer dos veces mas antes de que se acabe la línea. Esta expresión detectaría las líneas que contienen palabras (o conjuntos de palabras) triples.

- una expresiones regulares de un sólo carácter seguida de `\(entero\)` representa *entero* apariciones consecutivas de alguno de los caracteres aceptados por la expresiones regulares de un sólo carácter. Si va seguida de `\(entero,\)` representa un número mínimo de *entero* apariciones consecutivas. Si va seguida de `\(entero1, entero2\)` representa un mínimo de *entero1* y un máximo de *entero2* apariciones consecutivas. Los números enteros deben estar comprendidos entre 0 y 255.

- toda expresión que comienza con `^` indica que los caracteres buscados deben estar a comienzo de la línea

`^[a-z]*` selecciona las líneas que sólo contienen letras minúsculas

- toda expresión que termina con `$` indica que los caracteres buscados deben estar a final de la línea

`^[a-z]{3}\[^a-z]{3}` selecciona las líneas que comienzan con tres minúsculas, terminan con tres caracteres cualesquiera pero que no son minúsculas, y no tiene ningún otro carácter entre medio.

## 12.5 COMANDOS UTILES PARA TRABAJAR EN RED

### 12.6 Protocolos Internet (IP)

Cualquier comunicación entre dos sistemas distantes debe resolver los dos problemas siguientes: a) Designación de cada sistema (*addressing*), y b) Selección del camino a seguir por la comunicación (*routing*).

El protocolo IP define una dirección lógica para cada red local. La dirección de una máquina concreta se forma añadiendo a la dirección de la red el número que identifica a la máquina en esa red. La dirección completa tiene 32 bits, y se suele dar en la forma de 4 octetos separados por puntos (por ejemplo:

132.227.70.83). El número de octetos que designa a la red (izquierda) y a la máquina (derecha) es variable, dependiendo del tamaño de la red.

El camino de los mensajes (*routing*) se establece a través de ciertos ordenadores, llamados "pasarelas", que tienen la propiedad de pertenecer al menos a dos redes. Existen tablas que indican cual es la pasarela de la red local a través de la cual se puede acceder a otras redes.

Los mensajes enviados tienen un encabezamiento con las direcciones de los ordenadores "desde" y "hacia". Como los mensajes se suelen enviar fragmentados por problemas de tamaño, el encabezamiento lleva también la información necesaria para reconstruir el mensaje al llegar a su destino.

Las características principales de este nivel de comunicación entre ordenadores son: a) La conexión no es interactiva (se envía el mensaje y no se hace nada más), y b) No se garantiza ni la llegada, ni el orden de llegada, ni la no duplicación de los mensajes).

## 12.7 Denominación Simbólica de Sistemas *Internet*

En lugar de los números lógicos vistos previamente (del tipo 132.227.70.83), es posible utilizar nombres simbólicos tanto para las redes como para las máquinas. Estos nombres se estructuran de modo jerárquico, de modo similar al sistema de ficheros UNIX, tal y como puede verse en la Figura 36.

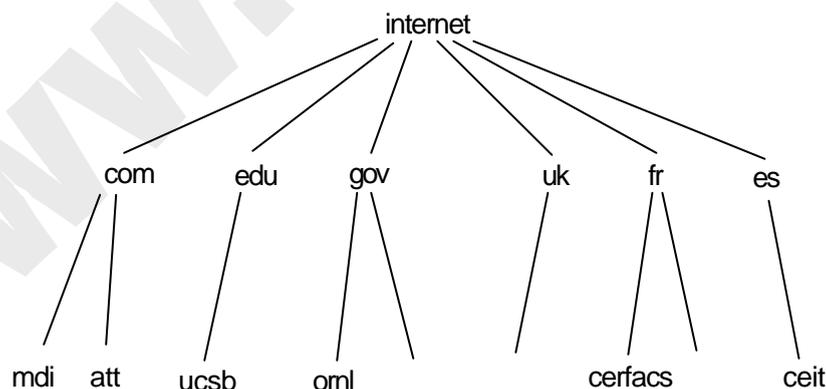


Figura 36

en la que por ejemplo, **com** indica empresas comerciales (como MDI o ATT), **edu** indica instituciones educativas americanas, **gov** instituciones gubernamentales, etc. A la derecha aparecen los indicativos de otros países, como el reino Unido, Francia y España. En la línea inferior aparecen las instituciones concretas de que se trate. Dentro de cada institución se pueden identificar las máquinas y los usuarios. El

comando **nslookup** permite hallar la dirección numérica **Internet** de un ordenador a partir de su nombre simbólico.

## 12.8 Comando telnet

Permite abrir una terminal virtual en un sistema distante. Este comando no requiere que los sistemas sean UNIX, y tiene la siguiente forma general:

```
telnet [direccion_host [puerta]]
```

donde la *direccion\_host* puede ser el número de **Internet**, el nombre en el fichero /etc/hosts o el nombre simbólico completo de la máquina. Si el sistema tiene varias puertas, hay que especificar por cual se desea hacer la conexión. Para salir de **telnet** se emplea el comando **quit** (o simplemente **q**).

## 12.9 Comando ftp

Permite la transferencia de ficheros entre sistemas distantes. Supone una conexión entre el sistema local y el sistema distante. Tiene la siguiente forma general:

```
ftp [-g] [-i] [-n] [-v] [direccion_host ]
```

Si se define la máquina distante *direccion\_host* se conecta con ella la máquina local. Si no, se entra en modo comando con el prompt **ftp>**. En ftp no se arranca un shell especial en la máquina distante. Las opciones tienen el siguiente significado:

- v** opción verbose
- i** opción no interactiva: no se pide confirmación para transferir cada fichero
- g** se inhibe el mecanismo de expansión de los nombres de ficheros (\*?)

Los comandos de **ftp** que afectan a la transmisión son los siguientes:

- ascii (binary)** transmisión en formato ASCII (por defecto) o binario.
- quit** fin de la conexión y de ftp
- close** fin de la conexión, sin salir de ftp
- glob** activación del mecanismo de expansión de ficheros
- open [dir\_host]** abrir conexión con el sistema *dir\_host*
- prompt** cambia de interactivo a no interactivo, y viceversa
- user username** identificación en máquina distante
- status** información sobre el proceso ftp
- verbose** activación de la opción verbose

Para ejecutar comandos sobre la máquina local se procede del siguiente modo:

- !comando** se ejecuta el comando en la máquina local
- lcd [dir]** cambio de directorio o vuelta al directorio base

Sobre la máquina distante, se dispone de los siguientes comandos:

- cd [dir]** cambio de directorio en la máquina distante
- delete filename** borrar un fichero
- dir [dir]** muestra el contenido del directorio actual.
- ls [dir]** al igual que el anterior, listado del contenido de un directorio
- mdelete [ref]** borrado de uno o más ficheros
- rmdir dir** borrado de un directorio

Finalmente, existen los siguientes comandos de transferencia de ficheros:

- append [ref\_local [ref\_distante]]** el fichero ref\_local se añade al fichero ref\_distante. Si ref\_distante no existe, se crea al hacer la transferencia
- put [ref\_local [ref\_distante]]** el fichero ref\_local se transmite con el nombre ref\_distante. Si se omite ref\_distante, se transmite con el mismo nombre
- send [ref\_local [ref\_distante]]** igual que el anterior
- mput ref\_local ...** similar a los anteriores, pero para varios ficheros
  
- get [ref\_distante [ref\_local ]]** el fichero ref\_distante se transmite con el nombre ref\_local.
- recv [ref\_distante [ref\_local ]]** igual que el anterior
- mget ref\_distante ...** similar a los anteriores, pero para varios ficheros