10029 ADMINISTRACIÓN DE SERVIDORES LINUX (UBUNTU/FEDORA)

Ramón M. Gómez Labrador

(ramongomez@us.es)

Junio de 2.010

Nota importante: El presente curso se oferta dentro del plan de formación para personal informático de la Universidad de Sevilla para el año 2010 y toda su documentación asociada está bajo licencia Creative Commons con reconocimiento (http://creativecommons.org/licenses/by/3.0/deed.es).

Ediciones previas de cursos sobre administración de Linux:

- 05-09 Administración Básica de Sistemas Linux, 3ª edición.
- 04-70 Administración Avanzada de Sistemas Linux, 3ª edición.
- 07048 Administración de Servidores Linux (Fedora/Ubuntu), 1ª edición.
- 08062 Administración de Servidores Linux (Ubuntu/Fedora), 2ª edición.
- 09026 Administración de Servidores Linux (Ubuntu/Fedora), 3ª edición.

10029 Administración de Servidores Linux (Ubuntu/Fedora)

Índice

1. Introducción.

Linux es un sistema operativo de la familia Unix, gratuito, creado mediante la política de "código abierto" [viii]. Estas características implican un gran ahorro en los costes de instalación de los equipos, pero también una mayor especialización por parte del personal informático.

En todo sistema Unix existe un usuario administrador (**root**), que controla el funcionamiento completo del sistema, tiene acceso universal y puede realizar cualquier operación con los datos y los dispositivos de la máquina.

Este curso se ofrece originalmente en el Plan de Formación para personal informático de la Universidad de Sevilla [i] y va dirigido principalmente a personas que, habiendo trabajado con el sistema operativo Linux y teniendo nociones esenciales de programación en BASH, se interesen por la realización de labores administrativas básicas en el sistema.

1.1. Tareas del administrador.

El administrador de cualquier tipo de servidor debe ser una persona especializada, que conozca lo mejor posible sus equipos, sus aplicaciones y sus usuarios; manteniéndose al día en los avances tecnológicos, en las revisiones y parches de los programas instalados y en las necesidades de su empresa.

1.1.1. Planificación y previsión de necesidades.

Una de las funciones principales en la administración de sistemas informáticos es la planificación detallada de las tareas de gestión, lo que puede evitar sorpresas desagradables en el momento de ejecutarlas.

El analista de sistemas tiene la obligación de asesorar al personal administrativo de su empresa sobre las necesidades tecnológicas en la adquisición de material informático, estimando los recursos que precisen los usuarios, en relación con las posibilidades económicas de la empresa.

Una vez recibido el equipo debe realizarse un plan de instalación, en el que se incluya, al menos la siguiente información:

- Documentación y estudio de los recursos disponibles.
- Previsión de posibles ampliaciones.
- Relleno de solicitud de alta en la red informática corporativa y activación de los parámetros de conexión.

- Documentación de necesidades del entorno de operación (SAI, aire acondicionado, etc.).
- Documentación sobre registro, configuración, instalación y actualización del sistema operativo, de las aplicaciones requeridos y de los programas propios, de acuerdo con los servicios que debe prestar el nuevo equipo.
- Creación y publicación de solicitudes de apertura y modificación de cuentas de usuarios, de instalación de programas, de mejora de recursos, etc.

1.1.2. Documentación.

El responsable del sistema se compromete a realizar también documentación interna para el Centro de Cálculo, que debe describir las siguientes necesidades:

- Registro actualizado de los usuarios y grupos del sistema.
- Políticas de utilización y permisos para cada grupo de usuarios.
- Descripción de los procedimientos comunes que deben ejecutar los operadores del sistema (copias de seguridad, gestión de cuentas, informes, etc.).
- Registro completo y actualizado de los cambios en la configuración del servidor (sistema operativo, aplicaciones, ficheros, etc.).
- Recogida periódica y archivado de datos sobre el rendimiento del sistema y de sus componentes.

1.1.3. Automatización.

El personal informático de una empresa ha de ejecutar periódicamente las funciones definidas en el plan de actuación. El programador necesita automatizar la mayoría de estos procedimientos repetitivos para evitar errores tipográficos o conceptuales, y para mejorar el tratamiento general de las aplicaciones.

En cada servidor deben automatizarse, al menos, las siguientes tareas:

- Comprobación del espacio libre en los discos.
- Gestión de cuentas de usuarios y revisión periódica de las cuotas de disco.
- Procedimientos para crear, comprobar y restaurar copias de seguridad, según el plan de actuación.
- Comprobación y registro del rendimiento general del sistema y de la red informática.

- Trabajos específicos (informes, gestión de servicios, creación de documentación, etc.).
- Creación de alertas de seguridad (comprobación de cambios, detección de intrusos, etc.).

1.1.4. Informar a los usuarios.

El administrador de sistema debe también mantener informados a sus usuarios y darles unas guías de operación y buen uso, lo que puede evitar errores provocados por desconocimiento.

También es necesario informar sobre los cambios que pueden afectar a cada grupo de usuarios, indicando la siguiente información [1]:

- La naturaleza de los cambios que van a realizarse en el sistema y su evolución temporal.
- Cuándo se realizará cada modificación.
- Qué resultados se esperan obtener con la operación y cuáles son los obtenidos.
- Tiempo estimado y tiempo real de la duración de la operación.
- Impacto posible sobre los usuarios (nueva configuración, parada del sistema, etc.).
- Información de contacto para recoger dudas y consultas.

Por otro lado, el encargado del sistema tiene la obligación de conocer profundamente el comportamiento general de sus usuarios, registrando sus consultas, sus sugerencias y los datos de rendimiento y utilización de recursos. Esto permite ofrecer una mejor calidad en los servicios ofertados.

1.1.5. Control de la seguridad del sistema.

Dependiendo del tipo de información tratada por el sistema, el administrador debe definir sus políticas de seguridad, tanto para el servidor, como para la red corporativa, ya que los usuarios tienen derecho a la privacidad e integridad de sus datos.

Deben ponerse los medios para evitar posibles ataques o fallos informáticos que afecten –o incluso paralicen– el funcionamiento normal de la máquina.

Nunca hay que tener la presunción de que un sistema es completamente seguro o de que sólo puede ser atacado desde fuera. Por ello, el *superusuario* debe realizar las siguientes operaciones:

- Activar y revisar los registros históricos de incidencias.

- Realizar revisiones periódicas sobre posibles cambios no deseados en el sistema.
- Instalar aplicaciones y dispositivos que protejan a los servidores y a la red informática (sistemas de detección de intrusos, cortafuegos, filtros, lectores de tarjetas de acceso, etc.)..

1.1.6. Previsión de fallos.

Por último, la empresa debe poner los medios físicos necesarios para prevenir y corregir los posibles fallos informáticos.

Por otra parte, los cambios ambientales (eléctricos, temperatura, humedad, ...) son algunos de los aspectos más importantes y costosos en la prevención de errores

. Debe hacerse hincapié en los siguientes temas:

- Tener una correcta instalación eléctrica, que evite caídas y subidas inesperadas de tensión, así como instalar sistemas de alimentación ininterrumpida (SAI) que protejan los servicios críticos de la empresa (armarios de comunicaciones, servidores, etc.).
- Tener un adecuado sistema de aire acondicionado, que filtre y regule la temperatura y la humedad del ambiente, sin que afecte a la salud de los operadores.
- Contar con un alumbrado adecuado, que no afecte al tendido eléctrico informático.
- Mantener una adecuada infraestructura en la red informática, con acceso cómodo y restringido a los dispositivos de comunicaciones.

Otras posibles causas de fallos más difíciles de prever son:

- Saturación o fallo de los recursos del sistema (procesadores, memoria, discos, etc.). Hay que sopesar la necesidad de solicitar la ampliación o sustitución de los componentes afectados.
- Fallos de programación, tanto en el S.O., como en las aplicaciones instaladas o en los programas propios. El administrador debe mantenerse informado sobre las actualizaciones y parches que tenga que instalar.
- Errores humanos del propio administrador, de los operadores, del servicio técnico o de los usuarios finales.

2. Usuarios y grupos.

Un usuario Unix representa tanto a una persona (**usuario real**) como a una entidad que gestiona algún servicio o aplicación (**usuario lógico o ficticio**) [2].

Todo usuario definido en el sistema se corresponde con un identificador único (**UID**) y con una **cuenta**, donde se almacenan sus datos personales en una zona de disco reservada.

Un **grupo** es una construcción lógica –con un nombre y un identificador (**GID**) únicos– usada para conjuntar varias cuentas en un propósito común [1], compartiendo los mismos permisos de acceso en algunos recursos. Cada cuenta debe estar incluida como mínimo en un grupo de usuarios, conocido como **grupo primario** o **grupo principal**.

2.1. Características generales de una cuenta.

Las características que definen la cuenta de un usuario son:

- Tiene un nombre y un identificador de usuario (UID) únicos en el sistema.
- Pertenece a un grupo principal.
- Puede pertenecer a otros grupos de usuarios.
- Puede definirse una información asociada con la persona propietaria de la cuenta.
- Tiene asociado un directorio personal para los datos del usuario.
- El usuario utiliza en su conexión un determinado intérprete de mandatos, donde podrá ejecutar sus aplicaciones y las utilidades del sistema operativo.
- Debe contar con una clave de acceso personal y difícil de averiguar por parte de un impostor.
- Tiene un perfil de entrada propio, donde se definen las características iniciales de su entorno de operación.
- Puede tener una fecha de caducidad.
- Pueden definirse cuotas de disco para cada sistema de archivos.
- Es posible contar con un sistema de auditoria que registre las operaciones realizadas por el usuario.

2.2. Ficheros del sistema.

Linux proporciona varios métodos para la definir los usuarios que pueden conectarse al sistema. Lo típico es definir localmente en cada servidor las cuentas de los usuarios y grupos, aunque también pueden usarse métodos externos de autentificación, que permiten que varias máquinas compartan las mismas definiciones para sus usuarios comunes.

La siguiente tabla muestra los ficheros del sistema involucrados en el proceso de definición de los usuarios locales.

Formato	Descripción
/etc/passwd	
Usuario:x:*UID*:*GID*:*Descrip*:*Direct*:*Shell* ...	Fichero principal de descripción de usuarios locales. Sus campos son: 1. Nombre de usuario. 2. No usado (antiguamente, clave). 3. Identificador de usuario (UID). 4. Identificador del grupo primario. 5. Descripción o nombre completo de la persona que representa dicho usuario. 6. Directorio personal. 7. Intérprete de mandatos.
/etc/shadow	
Usuario:*clave*:*F1*:*N1*:*N2*:*N3*:*N4*:*Caduc*: ...	Fichero oculto que incluye la codificación y las restricciones de las claves de acceso a las cuentas. Sus campos son: 1. Nombre de usuario. 2. Clave codificada. 3. Fecha del último cambio de clave. 4. Días hasta que la clave pueda ser cambiada. 5. Días para pedir otro cambio de clave. 6. Días para avisar del cambio de la clave.

	7. Días para deshabilitar la cuenta tras su caducidad. 8. Fecha de caducidad. 9. Reservado (normalmente ignorado). Nota: Las fechas se expresan como el nº de días desde el 1/1/1.970.
/etc/group	
Grupo:x:*GID*:*Usuarios* ...	Contiene la definición de los grupos de usuarios. Sus campos son: 1. Nombre del grupo. 2. No usado (antiguamente, clave del grupo). 3. Identificador del grupo (GID). 4. Lista de miembros (separada por comas).
/etc/gshadow	
Grupo:*Clave*:*Admins*:*Usuarios* ...	Fichero oculto y opcional que contiene las claves de grupos privados. Sus campos son: 1. Nombre del grupo. 2. Clave codificada (opcional). 3. Lista de usuarios administradores. 4. Lista de usuarios normales.

2.3. Usuarios y grupos predefinidos.

En todos los "dialectos" Unix existen algunos usuarios y grupos predefinidos por el sistema operativo, que se utilizan para la gestión y el control de los distintos servicios ofrecidos por el ordenador.

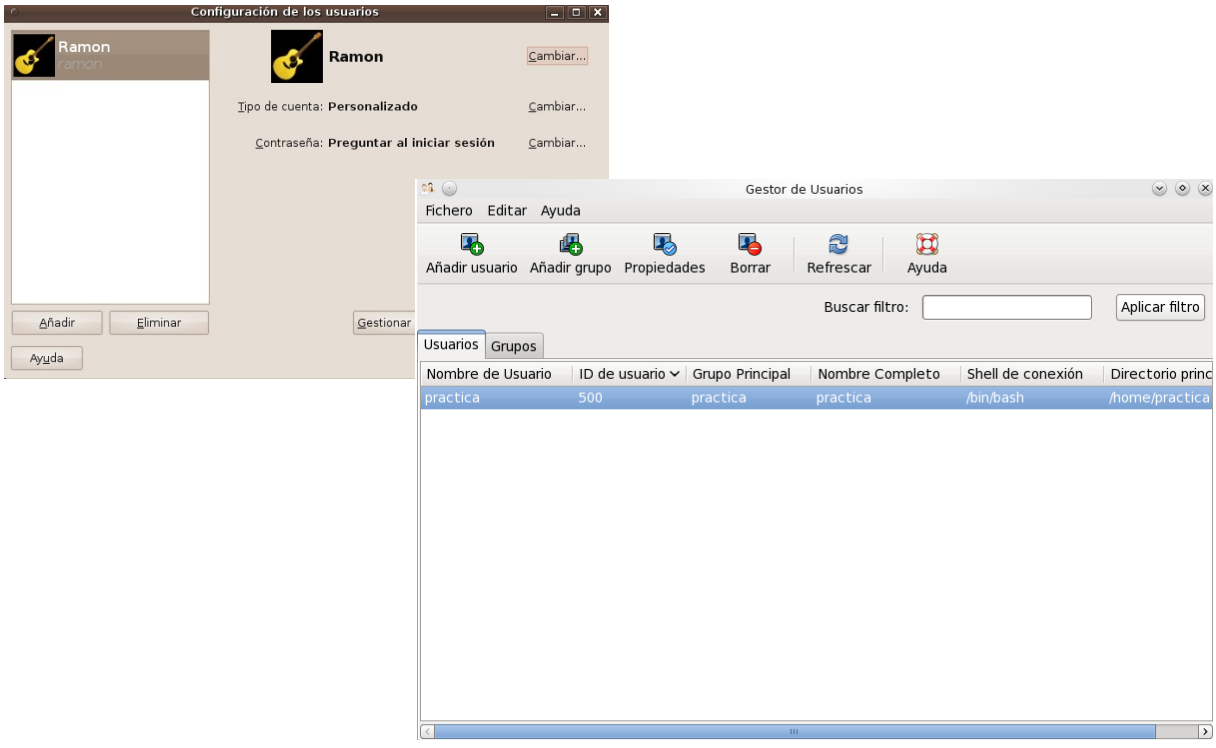
En especial el **usuario** `root` –con **UID 0**– es el administrador de la máquina, con un control total sobre el sistema. Existe también un **grupo** `root` –con **GID 0**– con características administrativas, al que pertenece el citado usuario.

Como ejemplo, la siguiente tabla lista algunos de los usuarios y grupos predefinidos en Fedora 13 [1] y en Ubuntu 10.04 Lucid [2], indicando también las posibles diferencias.

Usuario	UID [1]	UID [2]	Descripción
root	0	0	Administrador con control total.
bin	1	2	Propietario de las utilidades del sistema operativo.
daemon	2	1	Gestor de servicios generales.
adm sys	3	 3	Propietario de los archivos de registros históricos y administrativos.
lp	4	7	Administrador de los servicios de impresión.
nobody	99	65534	Gestor de servicios varios.
ftp	14	50	Controlador del acceso al árbol del servicio FTP anónimo..
sshd	74	123	Usuario ficticio gestor del servicio SSH.
apache www-data	48	 33	Propietario de los ficheros y directorios del servicio de hipertexto Apache.
squid proxy	23	 13	Controlador del servicio de representación Squid.
Grupo	**GID [1]**	**GID [2]**	**Descripción**
root	0	0	Administradores con control total.
bin	1	2	Binarios del sistema.
daemon	2	1	Servicios generales.
sys	3	3	Control del sistema.
adm	4	4	Ficheros históricos y administrativos.
tty	5	5	Acceso a la consola.
lp	7	7	Servicio de impresión.
kmem	9	15	Control de memoria del núcleo de Linux.

cdrom	11	24	Acceso a discos extraíbles (CD-ROM, DVD).
man	15	12	Páginas de manuales.
admin		123	Administradores que pueden ejecutar la orden sudo.
apache www-data	488	33	Servicio de hipertexto HTTP.
sshd ssh	484	111	Servicio de conexión segura SSH.
users	100	100	Usuarios normales.
nobody	99	65534	Control de servicios.
squid proxy	23	13	Servicio representante.

El gráfico siguiente muestra la ejecución de herramientas para gestión básica de usuarios como **users-admin** del entorno GNOME 2.30 bajo Ubuntu 10.04 Lucid (izquierda) y system-config-users de Fedora 13 (derecha).



Los usuarios ficticios, que gestionan los servicios ofrecidos por el ordenador, deben tener su cuenta deshabilitada para evitar una posible puerta de entrada para los intrusos. Esto se consigue bloqueando la clave de acceso y asignando `/sbin/nologin` como intérprete de mandatos de la cuenta.

2.3.1. El usuario `root` definido por defecto.

Ubuntu y Fedora establecen en sus programas de instalación distintas políticas para definir la forma de trabajar por defecto con la cuenta de superusuario (`root`).

Como se puede comprobar en el apartado anterior, en ambos casos la cuenta tiene el mismo nombre y los mismos parámetros de UID y GID. Sin embargo, el programa de instalación de Fedora pide establecer una clave para dicho usuario, mientras que el de Ubuntu no la solicita.

Ubuntu no permite conectarse directamente al sistema como `root` y sólo los usuarios que pertenecen al grupo `admin` pueden ejecutar órdenes con privilegios usando la orden **`sudo`** e introduciendo su propia clave.

La siguiente tabla muestra un resumen de las diferencias entre ambos sistemas operativos definiendo la cuenta `root`.

Característica	Ubuntu 10.04	Fedora 13
Clave de acceso	Sin clave	Definida en la instalación
Acceso al sistema	Sin acceso	Con acceso
Un usuario puede ejecutar órdenes como `root`	Orden **`sudo`** (miembros del grupo `admin`)	Orden **`sudo`** sin configurar
Un usuario puede ejecutar el intérprete de `root`	**`sudo su`** (miembros del grupo `admin`)	**`su`** (cualquier usuario)

2.4. Clave de acceso.

Como se ha indicado anteriormente, las claves de los usuarios locales de Linux se guardan codificadas en el fichero inaccesible `/etc/shadow`.

Los algoritmos de codificación de las claves son “de sentido único”, o sea que impiden la descodificación directa de las claves. Por lo tanto, cuando un usuario entra en el sistema, se le codifica la clave y se compara con la

clave válida encriptada. Si el resultado es correcto, el usuario puede conectarse.

Linux puede utilizar el algoritmo de codificación **Crypt**, usado en los antiguos sistemas Unix y llamado así por la función del lenguaje C que realiza los cálculos. Este método es inseguro porque usa claves de codificación débiles de 56 bits y las contraseñas sólo pueden tener un máximo de 8 caracteres.

Los nuevos Linux también soportan algoritmos de codificación más potentes como **MD5 o SHA**, mucho más robustos y que permiten claves más extensas y difíciles de averiguar. El algoritmo MD5 usa claves de 128 bits, mientras que SHA512 –por defecto en Fedora 13 y en Ubuntu 10.04– aumenta dicha longitud hasta los 512 bits.

La siguiente figura muestra la pestaña de opciones de la aplicación gráfica **`system-config-authentication`**, para gestión de autentificación en Fedora 13.

2.4.1. Restricciones para tener claves seguras.

El administrador debe recomendar a sus usuarios que creen claves que puedan resultar difíciles de averiguar para un pirata informático.

También debe hacer que el sistema cree dificultades al intruso, usando codificaciones complejas y creando restricciones que comprometan al usuario con la seguridad del sistema.

Todos los usuarios del sistema han de tener en cuenta las siguientes recomendaciones con sus claves:

- No usar palabras comunes o números asociados a la persona.

- No repetir las claves en distintas máquinas.
- Usar claves de 8 caracteres como mínimo, con al menos 2 caracteres no alfabéticos.
- No usar secuencias de teclado.
- Cambiar la clave periódicamente y no repetir claves anteriores.
- No dejar ni anotar la clave.
- Evitar que otra persona vea teclear la clave.

2.5. Permisos.

Uno de los elementos principales de la seguridad en Unix es el buen uso de los permisos para acceder a ficheros y directorios. Todo usuario –no sólo el administrador– debe tener claros los conceptos más básicos para evitar que otro usuario lea, modifique o incluso borre datos de interés [4].

El usuario administrador –al tener el control completo del sistema– también puede realizar operaciones sobre los ficheros y directorios de cualquier usuario (técnica que puede ser utilizada para evitar que un usuario pueda acceder a su propio directorio personal).

Este hecho hace imprescindible que los responsables de la máquina tengan especial cuidado cuando utilicen la cuenta del usuario **root**.

Los permisos de acceso se dividen principalmente en dos categorías:

- permisos normales,
- permisos especiales.

Por otro lado, los permisos también se subdividen en tres grupos:

- permisos para el propietario,
- permisos para su grupo de usuarios,
- permisos para el resto de usuarios del sistema,

Las **listas de control de acceso (ACL)** permiten asignar permisos de forma específica a conjuntos de usuarios y grupos.

2.5.1. Permisos normales.

Cada usuario tiene un nombre de conexión único en el ordenador y pertenecerá a uno o varios grupos de usuarios. El propietario de un fichero o directorio puede seleccionar qué permisos desea activar y cuales deshabilitar.

Para comprobarlo de manera más clara, tómese el primer grupo de valores obtenidos con el mandato **`ls -l`**, que permitirá observar los permisos. Estos 11 caracteres indican:

- 1 carácter mostrando el tipo: fichero (`-`), directorio (`d`), enlace (`l`), tubería (`p`), enlace simbólico (`l`), etc.
- 3 caracteres para los permisos del propietario.
- 3 caracteres para los permisos de otros usuarios del grupo.
- 3 caracteres para los permisos del resto de usuario.
- 1 carácter opcional que indica si hay definida una ACL.

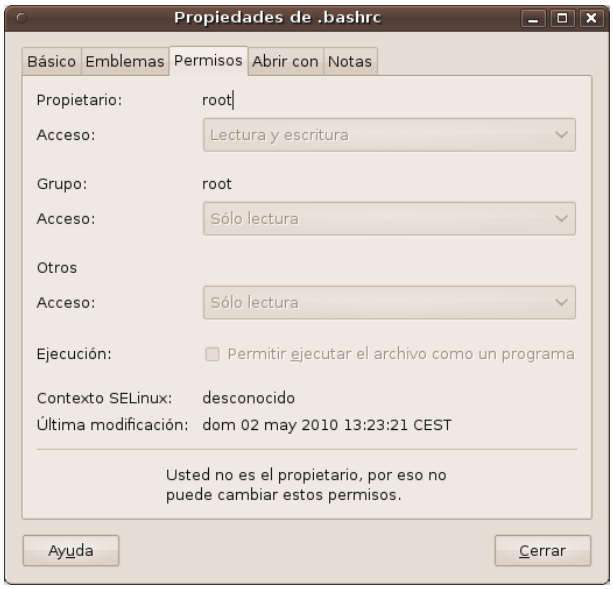
Según el tipo de entrada del directorio, los caracteres de permisos normales pueden variar de significado:

Ficheros:	**Lectura** (**r**): el usuario puede leer el fichero. **Escritura** (**w**): el usuario puede escribir en el fichero. **Ejecución** (**x**): el usuario puede ejecutar el fichero (siempre que sea un ejecutable o un guión de intérprete de mandatos).
Directorios:	**Lectura** (**r**): el usuario puede leer el contenido del directorio. **Escritura** (**w**): el usuario puede crear, modificar y borrar entradas del directorio. **Acceso** (**x**): el usuario puede acceder al directorio y puede usarlo como directorio actual (ejecutando la orden `cd`). Este permiso posibilita proteger cierta información de un directorio padre y, sin embargo, acceder a la información de los directorios hijos.

El ejemplo siguiente muestra la ejecución de una orden **`ls`** para ver el contenido y los permisos del directorio `/etc/skel` definido por defecto en Ubuntu 10.04 Lucid.

```
$ ls -al /etc/skel
total 32
drwxr-xr-x   2 root root  4096 2010-05-02 20:26 .
drwxr-xr-x 186 root root 12288 2010-05-30 13:49 ..
-rw-r--r--   1 root root   220 2008-05-12 20:49 .bash_logout
-rw-r--r--   1 root root  3103 2010-04-19 04:15 .bashrc
-rw-r--r--   1 root root   179 2010-03-26 13:31 examples.desktop
-rw-r--r--   1 root root   675 2008-05-12 20:49 .profile
```

Asimismo, el ejemplo muestra la pestaña “Permisos” del cuadro de propiedades del navegador de archivos propio del entorno GNOME 2.30 bajo Ubuntu 10.04, mostrando los datos del fichero `.bashrc` del directorio anterior.



La siguiente tabla muestra los permisos necesarios para poder ejecutar algunos mandatos [4].

Mandato	Permisos directorio origen	Permisos fichero	Permisos directorio destino
`cd`	**X**	No aplicable	No aplicable
`ls`	**R**	No aplicable	No aplicable
`mkdir`	**W, X**	No aplicable	No aplicable
`rmdir`	**W, X**	No aplicable	No aplicable
`cat`	**X**	**R**	No aplicable
`rm`	**W, X**	-	No aplicable
`cp`	**X**	**R**	**W, X**
`mv`	**W, X**	-	**W, X**

2.5.2. Permisos especiales.

Los permisos especiales complementan al conjunto de permisos normales, potencian la seguridad del sistema y se utilizan para soportar ciertas operaciones específicas.

Al igual que en el punto anterior, dependiendo del tipo de entrada del directorio, los caracteres de permisos especiales representados por **`ls -l`** son [4]:

Ficheros:	**Identificador de usuario activo** (**s** para el propietario): un programa ejecutable puede activar el identificador de usuario (**SUID**), lo cual permite que durante la ejecución del programa un usuario se convierta en el usuario propietario del fichero. Por ejemplo, el mandato **`passwd`** accede a ficheros que sólo puede modificar el usuario **`root`**. Dicho mandato tiene activo el SUID para que durante la ejecución del programa otro usuario sea por algún momento **`root`** y pueda cambiar su clave. Hay que tener especial cuidado con estos ejecutables, porque usuarios no autorizados pueden tomar privilegios. **Identificador de grupo activo** (**s** para el grupo): al igual que en el caso anterior, un programa ejecutable puede activar el identificador de grupo (**SGID**) para que un usuario pueda realizar operaciones propias del grupo al que pertenece el fichero. Por ejemplo, el mandato **`mail`** activa el SGID para que cualquier usuario pueda acceder a su buzón de correo sin posibilidad de leer correo de cualquier otro usuario.
Directorios:	**Directorio de intercambio** (**t** en el resto de usuarios): permite que en directorios compartidos los ficheros sólo puedan ser modificados por el propietario (suele usarse en directorios para ficheros temporales como `/tmp`. **Identificador de grupo activo** (**s** para el grupo): los ficheros que se creen en dicho directorio tendrán el mismo grupo que el del propio directorio, en vez del grupo del propietario.

El administrador debe catalogar todos los ficheros y directorios creados tras la instalación del sistema operativo o de cualquier aplicación, y que contengan permisos especiales. Periódicamente debe comprobar el estado de dichos archivos y verificar que no han sido modificados.

2.5.3. Notaciones simbólica y octal.

La orden **`chmod`** se utiliza para modificar los permisos de acceso descritos anteriormente y soporta dos tipos de notaciones: simbólica y numérica en formato octal.

La siguiente tabla muestra la forma de asignar permisos en ambas notaciones.

Permisos normales		Valor octal	Notación simbólica
Propietario:	Lectura	400	u+r
	Escritura	200	u+w
	Ejecución / Acceso	100	u+x
Grupo:	Lectura	40	g+r
	Escritura	20	g+w
	Ejecución / Acceso	10	g+x
Resto de usuarios:	Lectura	4	o+r
	Escritura	2	o+w
	Ejecución / Acceso	1	o+x
Permisos especiales		**Valor octal**	**Notación simbólica**
Propietario:	Usuario activo (SUID)	4000	u+s
Grupo:	Grupo activo (SGID)	2000	g+s
Resto de usuarios:	Directorio de intercambio	1000	+t

La notación simbólica se utiliza para añadir (+), quitar (-) o asignar (=) permisos agrupados según su tipo.

La notación numérica en formato octal sirve para asignar todos los permisos a la vez, aplicando una operación lógica O para obtener el resultado.

Véase un ejemplo. Si el usuario tiene permiso de modificación en el directorio y si es propietario de los archivos, se ejecutarán las siguientes modificaciones:

- A `fichero1` se le asignan los permisos de lectura y escritura para el propietario y el grupo asociado, y se le quitan (si existen) los de escritura y ejecución para otros usuarios.
- A `fichero2` se le asignan directamente los permisos de lectura y escritura para el propietario y de lectura para su grupo. El resto de usuarios no tiene ningún permiso.

```
$ chmod ug=rw,o-wx fichero1
$ chmod 640 fichero2
```

2.5.4. Listas de Control de Acceso (ACL)

La distribución básica de permisos de Linux es bastante rígida, sin embargo, las **Listas de Control de Acceso (ACL)** se implementan como una extensión al sistema de archivos para definir distintos conjuntos de permisos para usuarios y grupos de forma individualizada.

Una ACL consta de una lista de entradas en la que se especifica la asignación de los 3 permisos básicos de Unix: lectura (r), escritura (w) o ejecución/acceso (x), en este orden; la ausencia de un determinado permiso se denota por un guión (-). El siguiente cuadro muestra el formato genérico de una ACL.

```
[Tipo]:[Calificador]:ListaPermisos[,...]
```

La existencia de una ACL definida sobre un fichero se denota cuando el carácter número 11 de la salida de permisos de la orden **`ls -l`** es un signo más (+). La siguiente tabla describe las órdenes principales para gestión de ACL.

Mandato	Descripción
`getfacl`	Devuelve la lista de control de acceso a un fichero o directorio.
`setfacl`	Asigna, modifica o elimina una lista de control de acceso.

En el siguiente ejemplo, el propietario del fichero asigna permiso de lectura y escritura a un usuario determinado que no pertenece a su grupo.

```
$ ls -l fich1.txt
-rw-r-----  1 usu1 grupo1 11776 2010-01-27 19:20 fich1.txt
$ group usu2
usu2 : grupo2
$ setfacl -m user:usu2:rw fich1.txt
$ ls -l fich1.txt
-rw-r-----+ 1 usu1 grupo1 11776 2010-01-27 19:20 fich1.txt
$ getfacl fich1.txt
# file: fich1.txt
# owner: usu1
# group: grupo1
user::rw-
user:usu2:rw-
group::r--
other::r--
```

En este otro ejemplo se utiliza una orden compuesta para copiar la ACL de un fichero a otro.

```
getfacl fich1 | setfacl --set-file=- fich2
```

2.6. Configuración del entorno.

El intérprete de mandados de cada cuenta de usuario tiene un entorno de operación propio, en el que se incluyen una serie de variables de configuración.

El administrador del sistema asignará unas variables para el entorno de ejecución comunes a cada grupo de usuarios, o a todos ellos; mientras que cada usuario puede personalizar algunas de estas características en su perfil de entrada, añadiendo o modificando aquellas variables que el gestor no haya definido como de sólo lectura.

Para crear el entorno global, el administrador crea un perfil de entrada común para todos los usuarios (archivo `/etc/bashrc` en el caso de BASH), donde –entre otros cometidos– se definen las variables del sistema y se ejecutan los ficheros de configuración propios para cada aplicación.

Estos pequeños programas se sitúan en el subdirectorio `/etc/profile.d`; debiendo existir ficheros propios de los intérpretes de mandatos basados en el de Bourne (BSH, BASH, PDKSH, etc.), con extensión `.sh`, y otros para los basados en el intérprete C (CSH, TCSH, etc.), con extensión `.csh`.

El proceso de conexión del usuario se completa con la ejecución del perfil de entrada personal del usuario (archivo `~/.bash_profile` para BASH). Aunque el administrador debe suministrar un perfil válido, el usuario puede retocarlo a su conveniencia.

El lector puede dirigirse a la documentación de los cursos de formación sobre Programación Avanzada en Shell para obtener más información sobre las variables de entorno más importantes [3].

2.7. Gestión de cuentas.

Los sistemas Linux modernos (y los entornos de escritorio) ofrecen herramientas gráficas de gestión para realizar las tareas comunes de administración del sistema, incluyendo su propio conjunto de aplicaciones para la gestión usuarios y grupos.

Con este tipo de programas se pueden ejecutar las operaciones más sencillas de revisión y control, pero resultan bastante pobres para realizar una administración automatizada y avanzada de las cuentas de los usuarios.

El sistema operativo ofrece también una serie de mandatos de gestión, que deben ser usados para personalizar y automatizar el proceso de

creación, revisión y eliminación de usuarios y grupos. La siguiente tabla describe dichas funciones.

Mandato	Descripción
useradd	Crea una nueva cuenta de usuario.
usermod	Modifica los parámetros de una cuenta.
userdel	Borra una cuenta de usuario.
passwd	Modifica la clave de acceso a una cuenta.
chpasswd	Cambia la clave a varios usuarios usando un fichero de entrada de datos.
chage	Cambia las restricciones temporales de una cuenta.
chfn	Cambia la descripción del usuario.
chsh	Cambia el intérprete de mandatos de la cuenta.
groupadd	Crea un nuevo grupo de usuarios.
groupmod	Modifica los parámetros de un grupo de usuarios.
groupdel	Elimina un grupo de usuarios.
gpasswd	Cambia la clave de acceso a un grupo privado.

2.7.1. Planificación.

La gestión de las cuentas de los usuarios es uno de los aspectos más importantes dentro de las tareas administrativas, por ello deben planificarse detalladamente las características y las necesidades de los usuarios y de los grupos que vayan a darse de alta en el sistema.

Fundamentalmente, deben realizarse las siguientes operaciones previas antes de crear cualquier cuenta:

- Crear los distintos grupos de usuarios, uno para cada conjunto de tareas que vayan a ejecutar los usuarios, o uno por cada rol administrativo.
- Definir los parámetros globales del sistema, tales como: restricciones para la creación de claves, método principal de acceso, posibilidad de almacenamiento remoto de las cuentas, etc.
- Crear la estructura de directorios básica para las cuentas, separando los subdirectorios de cada grupo principal. Asignando los permisos

adecuados, puede evitarse que usuarios con menor privilegio accedan a zonas reservadas de otros grupos.

- Definir listas privadas donde el administrador pueda comprobar la identidad de cada usuario, almacenando los datos básicos de cada persona y de su cuenta asociada.
- Crear los programas para la gestión de las cuentas, generando ficheros de configuración que automaticen los procesos de creación, modificación, revisión, caducidad y borrado de usuarios.

2.7.2. Ejemplo: servidor de prácticas universitarias.

Para ilustrar el proceso de gestión de cuentas, la siguiente tabla describe en resumen una estructura que puede usarse en un servidor de prácticas universitarias, relativamente parecida a la existente en el Centro de Cálculo de la E.T.S. de Ingeniería Informática de la Universidad de Sevilla.

Restricciones generales para claves:	Las definidas anteriormente en este capítulo.
Creación de grupos de usuarios:	Crear grupos para administradores, alumnos normales, alumnos de proyectos fin de carrera y profesores. Un grupo para cada departamento. Definir grupos para alumnos por curso.
Estructura de directorios:	Directorio privado para el grupo de administradores. Directorio privado para profesores con subdirectorios privados para cada departamento. Directorios para alumnos normales agrupados por cursos y para alumnos de proyectos. Directorio para apuntes, con permisos de escritura para profesores y de lectura para alumnos.
Crear listas de usuarios:	Generar una lista distinta para cada grupo de usuarios..
Programas de gestión:	Creación de perfiles de configuración para los programas, donde se almacena información por defecto para cada tipo de usuarios y para la generación de los menús de selección. Creación interactiva de cuentas usando dichos perfiles. Creación automática de varias cuentas usando un fichero de datos de entrada.

	Comprobación de datos de usuarios; mostrando el contenido de la lista correspondiente, la entrada del fichero `/etc/passwd`, el directorio de la cuenta, la fecha de caducidad y la cuota de disco. Comprobación de concordancia entre los datos de las listas de usuarios y las cuentas creadas. Registro de cuotas de disco y comprobación semanal de su estado. Comprobación de la caducidad de las cuentas. Renovación automática de cuentas. Eliminación automática de cuentas caducadas. Borrado interactivo de cuentas y sus directorios. Registro de incidencias sobre bloqueo y desbloqueo de cuentas. Cambio automático de claves.

3. Sistemas de archivos.

La gestión adecuada del acceso a disco es otro de los aspectos importantes en el proceso de administración de sistemas operativos multiusuario y multitarea y es imprescindible mantener una estructura básica con un cierto nivel organizativo. El sistema operativo interactúa con los usuarios y las aplicaciones, y se hace necesario un modelo de seguridad dependiente de la forma en que se almacenan los ficheros en los dispositivos.

Un sistema de archivos puede verse desde dos categorías lógicas de ficheros [7]:

- Archivos locales no comprartibles o compartibles con otras máquinas.
- Archivos estáticos o variables.

Por lo tanto, un **sistema de archivos** es un subárbol de directorios con un directorio raíz –que debe tener unos permisos acordes con las necesidades de acceso a sus archivos–, una estructura lógica de almacenamiento y un punto de montaje adecuado en el árbol de directorios global del servidor.

3.1. Normas para la Jerarquía de Sistemas de Archivos (FHS).

Las **Normas para la Jerarquía de Sistemas de Archivos (FHS)** [v] describen un conjunto de reglas que permiten, tanto a los usuarios como a los programas, predecir la localización de los ficheros y directorios instalados en el sistema.

La siguiente tabla describe brevemente los subdirectorios de la jerarquía principal, ordenados alfabéticamente [7].

Directorio	Descripción
/bin	Binarios básicos para todos los usuarios del sistema.
/boot	Ficheros estáticos del cargador de arranque.
/dev	Sus entradas representan los dispositivos del sistema (conviene recordar que "*en Unix todo es un archivo*").
/etc	Configuración local del sistema.

/home	Cuentas de usuarios (si se define, debe ser un sistema de archivos independiente).
/lib	Bibliotecas compartidas del sistema y módulos fundamentales del núcleo.
/lib32 **/lib64**	Bibliotecas específicas para arquitecturas de 32 o 64 bits.
/media	Puntos de montaje para dispositivos extraíbles (disquete, CDs/DVDs, conexiones USB, etc.).
/mnt	Puntos de montaje para sistemas de archivos temporales.
/opt	Área compartida para paquetes de grandes aplicaciones (puede ser un sistema de archivos independiente con una jerarquía propia).
/proc	Sistema de archivos virtual con información sobre procesos y el núcleo.
/root	Cuenta del usuario administrador **root** (opcional).
/sbin	Binarios del sistema.
/srv	Datos de los servicios suministrados por el sistema.
/usr	Jerarquía secundaria similar a la principal, con información que puede ser compartida por otros ordenadores con acceso de sólo lectura (debe ser un sistema de archivos independiente en servidores).
/usr/local	Jerarquía para programas locales (debe ser un sistema de archivos independiente).
/tmp	Zona compartida para ficheros temporales.
/var	Información variable, incluyendo ficheros históricos, de estado, de bloqueos, de recuperación, de colas de trabajos, etc.

3.2. Discos y particiones.

Todos los sistemas Unix –y, por lo tanto, todos los "dialectos" Linux– utilizan ficheros de dispositivos para acceder a los recursos de la máquina, almacenados en el directorio `/dev`. Sin embargo, cada dialecto Unix tiene una notación diferente para identificar cada dispositivo de almacenamiento.

Tanto Fedora 13 como Ubuntu 10.04 Lucid identifican los ficheros controladores de particiones para discos sencillos con el siguiente formato:

- Tipo de dispositivo (`sd` para discos normales).
- Unidad (`a` para el dispositivo 1, `b` para el 2, etc.).
- Número de partición.

En caso de usar discos redundantes por *hardware*, el fichero del dispositivo se encuentra en un subdirectorio con el nombre del controlador RAID (por ejemplo, `cciss` para HP Smart Array) y su nombre tiene el siguiente formato:

- c*N* (nº de controlador, empezando por 0).
- d*N* (nº de disco, empezando por 0).
- p*N* (nº de partición, empezando por 1)..

Una **partición** es cada una de las subdivisiones que el gestor del sistema define en una unidad de disco del sistema, donde se almacena un determinado sistema de archivos o un espacio de paginación.

Siguiendo las normas descritas en el apartado anterior, el administrador debe definir los distintos sistemas de archivos de su sistema, creando particiones en cada disco, teniendo en cuenta los recursos disponibles y la utilización principal que los usuarios harán de ellos.

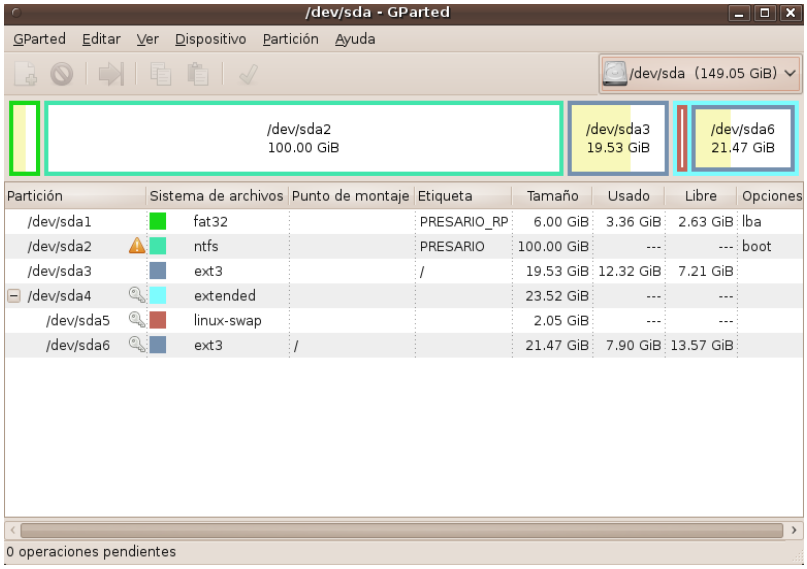
El proceso de crear los sistemas de archivos básicos suele realizarse durante la instalación de la máquina, aunque pueden añadirse y ampliarse posteriormente. La siguiente tabla define una distribución típica e indica algunas recomendaciones.

Sistema de archivos	Recomendaciones
`/`	Es necesario para trabajar, no tiene por qué ser de gran tamaño.
`/proc`	Es necesario para trabajar y debe ocupar entre la mitad y el doble de la memoria física, inversamente al tamaño de ésta.
`/usr`	Contiene el sistema operativo, su tamaño depende de los paquetes que deban instalarse y de las previsiones de ampliación.
`/tmp`	Espacio para ficheros temporales, depende del número de usuarios que se definirán y del espacio estimado para sus trabajos.
`/var`	Contará con ficheros que crecen, debe preverse un tamaño suficiente, pero sin desperdiciar el espacio de disco.
`/boot`	Puede usarse para almacenar el gestor de arranque; se usa cuando es necesario cargar algún controlador para localizar el sistema de archivos raíz.

/home	Cuentas de usuarios; puede ser recomendable usar un disco independiente, cuyo tamaño dependerá del número de usuarios y de la capacidad estimada de sus cuentas.
/usr/local	Debe tener un tamaño suficientemente grande para almacenar las utilidades y aplicaciones instaladas; es recomendable usar un disco independiente.

Aunque las distribuciones Linux suelen incluir herramientas gráficas que ayudan a gestionar el espacio de almacenamiento, tanto durante el periodo de instalación como de ejecución normal, pueden incluirse ciertas aplicaciones que tienen una apariencia independiente.

El siguiente gráfico describe la utilidad de creación de sistemas **GParted**, instalada en un sistema con 2 discos SATA. Se muestra el particionado del primer disco, con sistemas de archivos de Windwos (1 FAT y 1 NTFS) y de Linux (2 Ext3 y 1 de paginación).



3.3. Sistemas de archivos Ext3 y Ext4.

Linux soporta el montaje de distintos sistemas de archivos, tanto locales como remotos, ya que se ha programado una interfaz entre ellos y el núcleo, conocida como **Sistema de Archivos Virtual** (**VFS**).

El sistema de archivos más utilizado hasta hace algunos años en Linux era el conocido como **Sistema de Archivos Extendido 2** (**Ext2**), que aumentaba las prestaciones de la primera versión, pero que seguía presentando problemas ante una caída inesperada del sistema, ya que necesitaba un largo proceso de comprobación y corrección.

Las modernas distribuciones Linux usan el **Sistema de Archivos Extendido 3** (**Ext3**), el cual incluye las siguientes mejoras:

- El diario de registros es la característica más importante, que mejora los procesos de revisión de integridad, ya que sólo se requiere la comprobación de dicho diario.
- Soporta mayores niveles de integridad de datos para evitar la corrupción del sistema de archivos, permitiendo elegir el tipo y el nivel de protección.
- Mayor flujo y mayor velocidad de accesos repetidos a datos.
- Fácil transición entre ext2 y ext3, sin necesidad de volver a formatear las particiones.

El núcleo de Linux incluye soporte para el **Sistema de Archivos Extendido 4 (Ext4)** a partir de la versión 2.6.30; por lo tanto, Ext4 está incorporado por defecto en las instalaciones nuevas de Fedora 13 y de Ubuntu 10.04 Lucid.

El sistema de archivos Ext4 incluye las siguientes mejoras con respecto a Ext3 [ix]:

- Mayor tamaño del sistema de archivos (hasta 1 EB = 2^{20} TB).
- Sin restricciones en el número de subdirectorios.
- Mayor velocidad de tratamiento de ficheros grandes mediante "*extents*".
- Asignación previa de disco o asignación retardada.
- Comprobación del registro del sistema de archivos.
- Desfragmentación en directo sin necesidad de desmontar el sistema de archivos.
- Recuperación de ficheros borrados.
- Comprobaciones más rápidas del estado del sistema de archivos.
- Las marcas de tiempo cuentan con precisión de nanosegundos.
- Actualizable desde Ext3 (puede volverse a Ext3 perdiendo los "*extents*").

La configuración permanente de los sistemas de archivos montados en un servidor se define en el fichero `/etc/fstab`, incluyendo datos sobre el dispositivo origen, el punto de montaje, el tipo del sistema de archivos, así como el conjunto de opciones de montaje, depuración y comprobación de la consistencia de los datos.

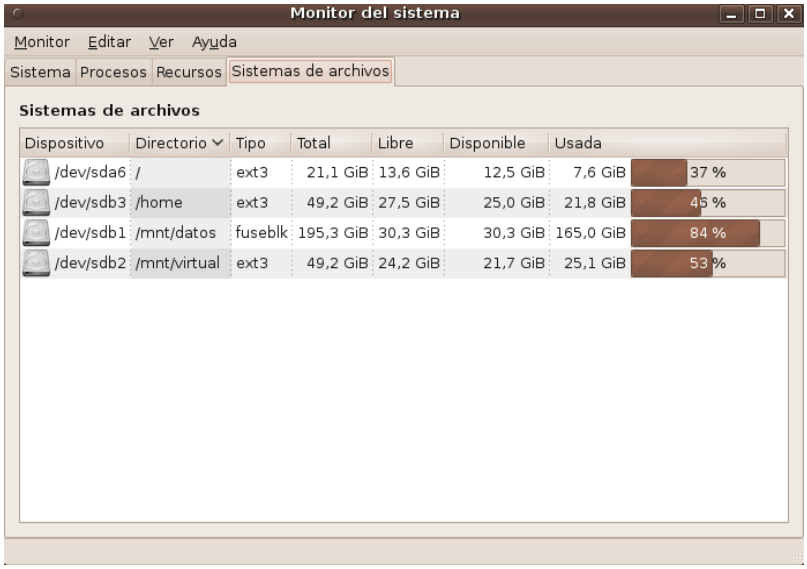
Las siguiente tabla describe el formato general del fichero de configuración `/etc/fstab`.

Formato	Descripción
/etc/fstab	
Etiq Montaje Tipo Opciones Volc NOrden ...	Fichero de descripción de sistemas de archivos. Sus campos son: 1. Etiqueta o UUID de la partición, dispositivo o directorio remoto. 2. Punto de montaje local. 3. Tipo de sistema de archivos (`ext3`, `ext4`, `swap`, `vfat`, `ntfs`, `nfs`, `cifs`, etc.). 4. Opciones de montaje (dependen del tipo de sistema de archivos). 5. Control de volcado automático de seguridad ante caídas del sistema. 6. Orden de comprobación de consistencia de datos durante el arranque del servidor (1 para `/`, incrementar en sistemas de archivos de distintos discos).

Asimismo, a continuación se describen brevemente los mandatos más habituales en la gestión de sistemas de archivos.

Mandato	Descripción
fdisk **sfdisk**	Manipulación de la tabla de particiones de un disco.
parted	Gestión de particiones y sistemas de archivos.
mkfs	Formatea una nueva partición.
mount	Monta un sistema de archivos en el árbol global de directorios.
umount	Desmonta un sistema de archivos.
tune2fs	Conversor entre sistemas de archivos **ext2**, **ext3** y **ext4**.

El siguiente gráfico muestra un ejemplo de la ejecución de la pestaña “Sistemas de archivos” de la aplicación **`gnome-system-montor`**, incluida en GNOME 2.30 bajo Ubuntu 10.04.



3.4. Paginación y procesos.

3.4.1. Espacios de paginación.

Un sistema operativo multiusuario y multitarea como Linux necesita una gran cantidad de memoria física para poder ejecutar todos los procesos. Los **espacios de paginación** son particiones de disco que permiten ampliar virtualmente la memoria del sistema, guardando el estado de los procesos que en un determinado momento están a la espera de ser ejecutados, si la memoria física está agotada.

Los factores principales que deben determinar el tamaño del espacio toral de paginación son:

- La cantidad de memoria y de disco del sistema.
- El número de usuarios que tendrán acceso a la máquina.
- El número previsto de procesos/usuario.
- El número de servicios activos en el sistema.
- El número estimado de clientes/servicio.

Debido al crecimiento de la memoria en los nuevos servidores, la regla general es utilizar para paginación entre la mitad y el doble de la memoria física instalada. Ante casos de necesidad, el administrador puede ampliar la cantidad de paginación usando ficheros de disco que pueden ser posteriormente eliminados.

La siguiente tabla describe las órdenes Linux usadas para manipular los espacios de paginación.

Mandato	Descripción
fdisk	Gestor de discos usado para definir particiones.
mkswap	Crea particiones o ficheros de paginación.
swapon	Activa una partición o un fichero de paginación.
swapoff	Desactiva una partición o un fichero de paginación.

3.4.2. Sistemas de archivos virtuales /proc y /sys.

Los sistemas de archivos virtuales montados en `/proc` y en `/sys` están almacenados en memoria y contienen una jerarquía de ficheros y directorios especiales que mantienen el estado actual del núcleo del sistema Linux, recopilando información sobre los dispositivos y los procesos en ejecución.

El siguiente cuadro muestra el formato típico de la definición de ambos sistemas de archivos en el fichero `/etc/fstab`.

```
sysfs          /sys              sysfs    defaults        0 0
proc           /proc             proc     defaults        0 0
```

La mayoría de los ficheros virtuales de `/proc` aparecen con longitud 0, aunque pueden ser revisados como si fueran archivos de texto, algunos de ellos con gran cantidad de información [2].

En `/proc` hay una serie de directorios especiales que describen el estado actual de cada proceso en ejecución –denotados por el identificador del proceso (PID)–, incluyendo datos como: la línea de la orden ejecutada, los directorio raíz y de trabajo del proceso, estados de la memoria, de ejecución y de uso de los procesadores, las variables de entorno, etc.

Otros ficheros y directorios de interés son los que informan sobre procesadores, memoria, interrupciones, dispositivos, particiones, puntos de montaje, módulos del núcleo, parámetros de red, etc.

El directorio especial `/proc/sys` contiene ficheros que sólo pueden ser modificados por el administrador para realizar cambios de configuración en el núcleo, habilitando o desactivando ciertas características operativas.

Debe tenerse gran precaución en la modificación de los archivos virtuales de `/proc/sys`.

La siguiente tabla describe el contenido de dicho directorio `/proc/sys`.

Directorio	Descripción
/proc/sys/dev	Información sobre dispositivos especiales (CD-ROM, discos RAID, etc.).
/proc/sys/fs	Parámetros de sistemas de archivos (límites de ficheros e i-nodos abiertos, cuotas, etc.).
/proc/sys/kenrel	Configuración del núcleo (contabilidad de procesos, nombre del sistema, parada por consola, módulos, colas de mensajes, etc.).
/proc/sys/net	Parámetros de conexión para cada tipo de red (IPv4, IPv6, Token Ring, local (loopback), etc.).
/proc/sys/vm	Configuración de la memoria virtual (páginas usadas, tamaño mínimo que la memoria debe quedar libre, etc.).

La información completa sobre el sistema de archivos virtual /proc puede encontrarse en el paquete del código fuente del núcleo, que suele encontrarse en el subdirectorio /usr/src/*VersiónNúcleo*/Documentation.

Por su parte, el núcleo de Linux utiliza el sistema de archivos /sys para informar a las aplicaciones de usuario respecto de los objetos gestionados por el Kernel y las relaciones de dependencia entre ellos.

La estructura principal de /sys agrupa información sobre las clases de dispositivos registrados, los buses físicos, los dispositivos conectados, los controladores y los módulos del núcleo.

3.5. Discos redundantes (RAID).

La **Matriz Redundante de Discos Independientes** (**RAID**) representa un conjunto de técnicas válidas para ahorrar costes o mejorar las prestaciones y la seguridad del acceso al almacenamiento masivo, combinando múltiples discos en un único dispositivo lógico [2].

El concepto principal de RAID es dividir los datos en ciertos trozos y distribuirlos en los dispositivos de la matriz, según el nivel de necesidad. Durante el proceso de lectura se sigue un algoritmo inverso de reconstrucción.

Las principales características del uso de discos en RAID son:

- Aumentar la velocidad de acceso a los datos.
- Incrementar la capacidad de almacenamiento, combinando discos de menor capacidad en un único disco lógico mayor.

- Mejorar la tolerancia a fallos de los discos.

Existen dispositivos y controladores preparados para realizar técnicas RAID en su propio *hardware*, lo que aumenta las prestaciones y el precio final de la máquina. Los nuevos sistemas operativos son aptos para realizar este cometido bajo *software*.

La siguiente tabla describe los niveles RAID más usados.

RAID 0:	Los datos se dividen en bandas, escribiendo cada una de ellas en un disco. Se mejora las prestaciones de acceso. La capacidad total es la suma de las capacidades de cada disco.
RAID 1:	Los datos se almacenan en espejo, repitiendo la misma escritura en cada disco. Se incrementa la seguridad y la tolerancia a fallos del sistema, porque puede sustituirse un disco defectuoso sin afectar al funcionamiento de la máquina. La capacidad total corresponde a la de cualquier disco (todos deben ser iguales).
RAID 5:	Se usan más de 2 discos para distribuir los trozos de datos y sus paridades. Cada disco contiene una banda de datos y la paridad de las bandas de otros datos. Se incrementan la seguridad, las prestaciones y los costes. La capacidad total es, aproximadamente, la suma total de la capacidad de los discos menos 1.
RAID 6:	Es un RAID 5 con 2 bandas de comprobación. Aumenta aún más la seguridad del sistema y sus costes. Se disminuye en algo la capacidad total con respecto al RAID 5.
RAID lineal:	Los discos se agrupan secuencialmente para formar un disco lógico mayor. No se incrementan ni las prestaciones ni la seguridad, sólo la capacidad.

Si las particiones que vayan a utilizarse para el RAID por *software* van a montarse sobre sistemas de archivos esenciales para Linux, deben definirse durante el proceso de instalación del sistema operativo. Utilidades como **Disk Druid** (usada en Fedora y Red Hat Enterprise) permiten definir particiones RAID, asociarles el nivel de redundancia y generar el disco lógico (**metadispositivo**).

La definición de la matriz se encuentra en el fichero `/etc/raidtab`. El siguiente ejemplo muestra la configuración de un metadispositivo `md0` de tipo RAID 1 (espejo) formado por las particiones `sda1` y `sdb1` de 2 discos, que contienen un sistema de archivos de tipo Ext3 montado sobre el directorio raíz.

```
$ cat /etc/raidtab
raiddev             /dev/md0
raid-level                  1
nr-raid-disks               2
chunk-size                  64k
persistent-superblock       1
nr-spare-disks              0
    device          /dev/sda1
    raid-disk       0
    device          /dev/sdb1
    raid-disk       1
$ df -h /dev/md0
S.ficheros          Tamaño Usado   Disp Uso% Montado en
/dev/md0               40G 21,5G  19,5G  51% /
```

3.6. Volúmenes lógicos.

Los volúmenes lógicos son técnicas de gestión de almacenamiento disponibles a partir de la versión 2.4 del núcleo de Linux –heredadas del sistema operativo AIX, el dialecto Unix de IBM– que permiten redimensionar las particiones y distribuirlas en varios discos.

En algunas distribuciones de Linux puede existir la restricción impuesta por el **Gestor de Volúmenes Lógicos** (**LVM**) de que el directorio `/boot` deba encontrarse en una partición real y no formar parte de ningún volumen lógico.

Es obligatorio definir los volúmenes lógicos en el proceso de instalación cuando éstos vayan a almacenar sistemas de archivos propios del sistema.

El Gestor de Volúmenes Lógicos consta de 3 elementos fundamentales:

Volumen físico:	estructura que representa a un disco físico.
Volumen lógico:	estructura equivalente a un sistema de archivos Linux.
Grupo de volúmenes:	conjunto de varios volúmenes lógicos que pueden almacenarse en varios volúmenes físicos. Así, un disco puede contener varios sistemas de archivos y un sistema de archivos puede estar grabado en varios discos.

El instalador del sistema debe seguir los siguientes pasos:

- Si la distribución de Linux es antigua, crear una partición normal de tipo Ext3 para el directorio `/boot`, ya sea incluido en el directorio raíz o en una partición propia.
- Definir un volumen físico en cada disco.
- Crear los grupos de volúmenes conjuntando adecuadamente los volúmenes físicos.
- Definir los volúmenes lógicos de cada grupo de volúmenes, asignando para cada uno de ellos su tamaño inicial y su punto de montaje.

Es recomendable dejar algún espacio sin asignar para poder ampliar las particiones que lo necesiten.

El sistema incluye una gran variedad de mandatos para gestionar cada uno de los componentes del gestor de volúmenes lógicos. La siguiente tabla describe la mayoría de estas instrucción según su función.

Operación	Volumen físico	Grupo de volúmenes	Volumen lógico
Crear	**pvcreate**	**vgcreate**	**lvcreate**
Eliminar	**pvremove**	**vgremove**	**lvremove**
Comprobar estado	**pvscan**	**vgscan**	**lvscan**
Cambiar tamaño	**pvresize**	**vgresize** **vgextend** **vgreduce**	**lvresize** **lvextend** **lvreduce**
Mostrar información	**pvs**	**vgs**	**lvs**
Mostrar atributos	**pvdisplay**	**vgdisplay**	**lvdisplay**

El siguiente cuadro muestra un ejemplo real usado para definir un grupo de volúmenes `vg0` con 2 discos que usan un controlador RAID por *hardware*, en donde se definirán 3 volúmenes lógicos; posteriormente, el administrador podrá usar cada uno de ellos para montar los sistemas de archivos del servidor.

```
# pvcreate /dev/cciss/c1d0 /dev/cciss/c1d1
# pvs
  PV             VG   Fmt  Attr PSize   PFree
  /dev/cciss/c1d0      lvm2 --     1,36T   1,36T
  /dev/cciss/c1d1      lvm2 --   698,56G 698,56G
# vgcreate vg0 /dev/cciss/c1d0 /dev/cciss/c1d1
# vgs
  VG   #PV #LV #SN Attr   VSize VFree
  vg0    2   0   0 wz--n- 2,05T 2,05T
```

```
# lvcreate -L 300G vg0
# lvcreate -L 700G vg0
# lvcreate -L 700G vg0
# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Log Copy%
  lvol0 vg0  -wi-a- 300,00G
  lvol1 vg0  -wi-a- 700,00G
  lvol2 vg0  -wi-a- 700,00G
# vgs
  VG   #PV #LV #SN Attr   VSize VFree
  vg0    2   3   0 wz--n- 2,05T 395,69G
```

3.7. Sistemas de archivos remotos.

La conexión remota a otros ordenadores supone una gran ventaja en el proceso de compartir información. Los sistemas de archivos remotos permiten almacenar la información en un único nodo central y hacerla accesible a los distintos clientes, posibilitando la movilidad del usuario.

Para finalizar este capítulo van a describirse los sistemas de archivos remotos más utilizados actualmente.

3.7.1. NFS.

El **Sistema de Archivos en Red** (**NFS**) fue creado por Sun Microsystems para SunOS –su dialecto Unix–, usando las técnicas de **Llamadas a Procedimientos Remotos** (**RPC**). NFS permite acceder a los archivos en nodos remotos exactamente en la misma manera que si fueran locales, de un modo completamente transparente al cliente e independientemente de la arquitectura del servidor [6].

IETF [xi] especifica en su RFC 3530 la versión 4 de NFS (**NFSv4**), redefiniendo completamente el protocolo e incluyendo mejoras como bloqueo de uso de ficheros, negociación de seguridad, ACLs, interoperabilidad entre plataformas, internacionalización, etc.

La siguiente tabla describe los servicios que deben activarse en los ordenadores servidor y cliente NFS.

Servicio	Descripción
portmap	Servicio de control principal de RPC.
rpc.mountd	Control de montaje del cliente NFS.
rpc.nfsd	Servidor NFS.

rpc.statd	Monitor del Estado de la Red (NSM), que notifica el reinicio del servidor NFS.
rpc.rquotad	Provee información de cuotas para usuarios remotos.

El fichero `/etc/exports` contiene la configuración NFS en el servidor. La siguiente tabla describe el formato de las líneas del fichero, una para cada directorio exportado.

Formato	Descripción
/etc/exports	
Directorio Cliente(*Opciones*)	Fichero principal que describe los directorios que pueden exportarse por NFS. Sus campos son: a) Directorio local a exportar. b) Nombre o IP del cliente (soporta comodines en nombre y en dominios). c) Opciones de exportación: sólo lectura (`ro`), lectura/escritura (`rw`), evitar acceso privilegiado para el **root** del cliente (`root_squiah`), acceso privilegiado para **root** (`no_root_squash`), etc.

El cliente NFS puede configurar la importación de directorios en su fichero `/etc/fstab` o montarlo directamente con la orden **mount**.

```
# mount –t nfs4 Servidor:Directorio PuntoMontaje [Opciones]
```

3.7.2. SMB/CIFS.

El **Sistema de Archivos Común para Internet** (**CIFS**) provee una serie de mecanismos abiertos e independientes de la plataforma utilizada, para que sistemas clientes soliciten servicios de ficheros a otras máquinas a través de la red. Este protocolo es una implementación del conocido como **Bloque de Mensajes del Servidor** (**SMB**), usado principalmente por ordenadores con Windows [12].

Microsoft ha redefinido nuevos dialectos del protocolo (SMB2 lanzado con Windows Vista y SMB2.1 con Windows 7), para mejorar el rendimiento y reducir la complejidad de las comunicaciones.

Las características principales de CIFS son:

- Acceso a ficheros, permitiendo compartir información en lectura y escritura.
- Acceso bloqueado y desbloqueado tanto a ficheros como a registros.
- Notificación de cambios en ficheros y directorios.
- Inclusión de atributos extendidos.
- Independencia del protocolo de resolución de nombres.

Las **Extensiones de CIFS para UNIX** son normas de reciente creación y sólo están implementadas en las versiones 2.6 de los servicios de ficheros del núcleo de Linux, mientras que los antiguos necesitan ser recompilados o generar un módulo propio para la gestión de clientes CIFS, aunque soportan el montaje de sistemas de archivos SMBFS.

El servidor de ficheros puede ser una máquina con sistema operativo Windows (a partir de NT) o con Linux y el servicio **Samba** activado. En ambos casos, deben ser configurados los recursos que van a ser exportados.

Cada distribución de Linux incluye una serie de paquetes con las herramientas básicas para el control de sistemas de archivos CIFS/SMB, los clientes para acceso a los recursos o el servidor de ficheros Samba.

Ubuntu 10.04 Lucid distribuye la rama 3.4 de Samba, que incorpora compatibilidad inicial con la futura versión de Samba 4; mientras que Fedora 13 viene con la reversión 3.5, la cual incluye soporte experimental para SMB2.

La próxima tabla muestra los mandatos usados por el cliente Samba.

Mandato	Descripción
smbclient	Cliente Samba con interfaz similar al cliente FTP.
smbpasswd	Permite cambiar la clave remota del usuario..
smbcquotas	Gestiona las cuotas en recursos NTFS.
smbcacls	Gestiona la lista de control de accesos (ACL) a los ficheros.
smbspool	Envía un fichero a una cola de impresión remota.
net	Herramienta de administración de Samba y de servidores remotos.
pdbedit	Herramienta de de gestión la base de datos de usuarios de Samba.
findsmb	Lista las máquinas que responden a una petición SMB en una subred.
mount.cifs	Montador de sistemas de archivos CIFS.
umount	Desmontador general de sistemas de archivos.

El montaje de un sistema de archivos CIFS requiere autentificación mediante usuario y clave. El método más seguro es indicar en la orden de montaje un archivo donde se incluyan las credenciales del usuario (usuario, contraseña y dominio opcional de autentificación), con el siguiente el formato:

```
Username = UsuarioRemoto
Password = Clave
Domain = Dominio
```

El formato para montar un sistema de archivos CIFS es el siguiente:

```
mount –t cifs //Servidor/Recurso PuntoMontaje \
      –o credentials=FichCredenciales[,Opción=Valor,...]
```

4. Configuración de la red.

La red informática es el medio por el cual el servidor puede comunicarse con los usuarios y con otras máquinas, tanto servidores como clientes, permitiendo el intercambio masivo de información entre ordenadores.

De acuerdo con la planificación efectuada, la empresa debe contar con una infraestructura adecuada para el intercambio de datos. Asimismo, los dispositivos de los servidores deben cumplir las necesidades previstas, ofreciendo un ancho de banda y una capacidad de procesamiento adecuados.

Existe una gran variedad de tipos de redes y protocolos de comunicaciones, sin embargo, este capítulo se centra en redes Ethernet con protocolos TCP/IP, los más usados en la conexión a Internet y en redes privadas.

4.1. Interfaces de red.

El ordenador necesita un dispositivo –conocido como **tarjeta de red**– que le permita conectarse a cada una de las subredes que tenga directamente a su disposición.

El sistema operativo Linux puede trabajar con una gran variedad de tipos de máquinas y periféricos. Para normalizar el acceso a la red, el sistema dispone de una serie de funciones básicas. El conjunto de estas funciones usadas en una arquitectura de comunicaciones determinada, se conoce como **interfaz de red**.

Por último, la interfaz de red dialoga con el dispositivo físico mediante un módulo específico del núcleo denominado **controlador de red**.

Las modernas versiones de Linux detectan automáticamente las tarjetas de red, cargan los módulos adecuados del núcleo y asignan los interfaces de red por defecto. El administrador puede establecer los parámetros de conexión durante el proceso de instalación del sistema.

Linux establece una nomenclatura para cada tipo de interfaz de red, añadiendo un número de orden para cada conector del mismo tipo (empezando por el número 0). La siguiente tabla describe la nomenclatura usada por Red Hat para los principales interfaces de red.

Interfaz	Descripción
`lo`	Interfaz virtual para pruebas (tiene asignada la dirección IP 127.0.0.1).
`eth`	Dispositivos Ethernet (también puede definir dispositivos ADSL y Ethernet .).

`wlan`	Dispositivos Ethernet inalámbricos.
`tr`	Redes en anillo de tipo Token Ring.
`ppp`	Conexión mediante módem o RDSI.
`hdi`	Dispositivo BlueTooth.

Cada dispositivo de red cuenta con una dirección física de acceso al medio (**dirección MAC**) única y diferente, asignada por el fabricante. Sin embargo, durante el proceso de activado del interfaz de red deben asignarse sus parámetros de conexión.

La dirección MAC de una tarjeta Ethernet está formada por 48 bits representados en 6 campos con 2 dígitos hexadecimales cada uno.

4.2. TCP/IP.

El protocolo de comunicaciones **TCP/IP** (*Transmision Control Protocol/Internet Protocol*) permite la localización y comunicación de todo tipo de máquinas conectadas a Internet. TCP/IP está constituido por un conjunto de protocolos basado en capas [4]:

- La capa de red –equivalente al nivel 3 de la norma OSI–, que establece el camino óptimo que deben seguir los paquetes de información que comunican varias máquinas. Utiliza el **Protocolo de Internet** (**IP**).
- La capa de transporte –equivalente al nivel 4 de la pila de protocolos OSI–, que permite establecer una conexión entre nodos de la red. Existen 2 protocolos de transporte: el **Protocolo para el Control de la Transmisión** (**TCP**) –que realiza una comunicación síncrona y segura con recuperación de datos en caso de error– y el **Protocolo de Datagramas del Usuario** (**UDP**) –que permite una comunicación asíncrona basada en paquetes denominados **datagramas**.

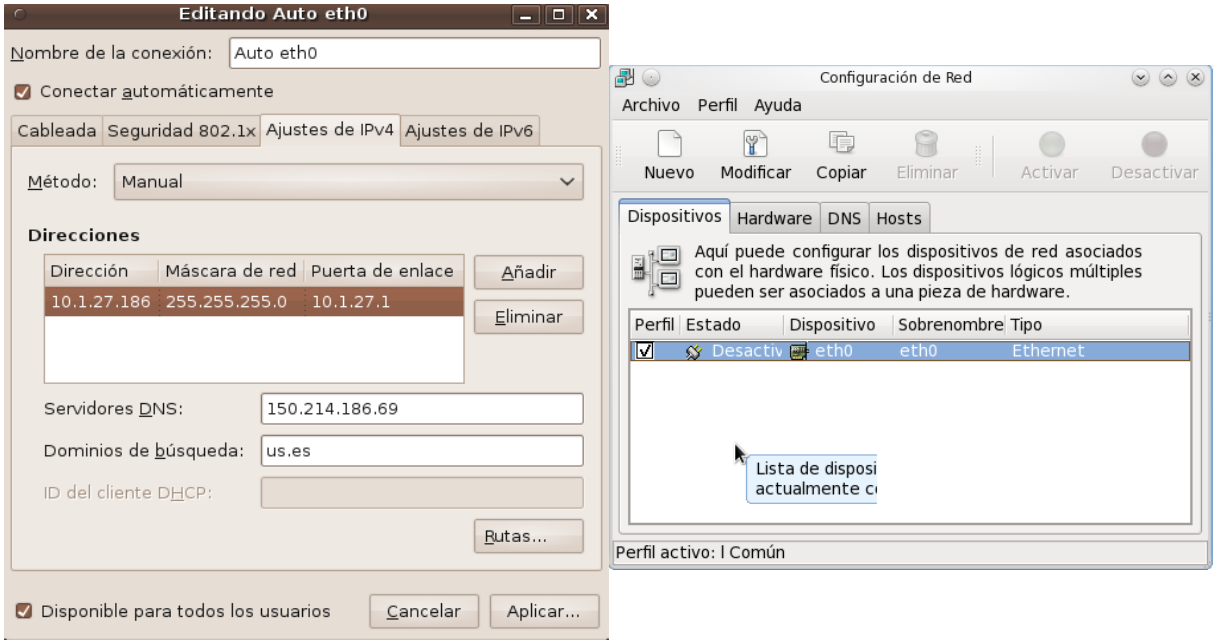
El conjunto de protocolos TCP/IP establece un mecanismo basado en direcciones y nombres que permite localizar inequívocamente cada máquina conectada a Internet. Las equivalencias entre direcciones IP y nombres de máquinas son realizadas por ordenadores especiales que atiendes las consultas mediante el protocolo conocido como **Servicio de Nombres de Dominios** (**DNS**).

El administrador del sistema tiene que establecer los parámetros para cada interfaz de red del sistema, bien mediante ficheros de configuración locales, bien generados por un servidor DHCP remoto, que puede asignar

los valores estática o dinámicamente. En ambos casos, deben especificarse los aspectos descritos en la siguiente tabla.

Dirección IP del interfaz:	Dirección única y diferenciada en toda Internet o en la red privada, formada por 32 bits en IPv4 o por 128 bits en IPv6.
Máscara de red:	Especifica mediante una operación lógica Y la porción de bits de la dirección IP común a todas las máquinas de la misma subred.
Dirección de difusión de la red:	Usada para enviar paquetes de información a todos los dispositivos de la misma subred.
Nombre del nodo y nombre del dominio de red:	Ambos valores en conjunto describen fácil y unívocamente una determinada máquina en toda Internet o en la red privada.
Direcciones de los servidores de nombres:	Servidores encargados de la resolución de nombres en Internet mediante el protocolo DNS. No suele usarse en redes privadas.

La siguiente figura muestra la pantalla de opciones de configuración de **NetworkManager**, incluida en GNOME 2.30 de Ubuntu 10.04 (izquierda), y la pantalla principal de **`system-config-network`**, suministrada con Fedora 13 (derecha); herramientas que pueden utilizarse en la configuración básica de las interfaces de red.



4.3. Configuración de la red.

Para terminar la instalación básica de la red, el responsable del sistema debe revisar –y en algunos casos modificar– los ficheros de configuración de los servicios esenciales del sistema. La siguiente tabla describe los formatos de estos ficheros.

/etc/sysconfig/network	
Descripción:	Usado en Fedora para establecer los valores de las variables básicas para el servicio de red (nombre, dominio, dirección del *encaminador*, etc.
Formato:	*Variable*=*Valor* ...
/etc/sysconfig/network-scripts/ifcfg-*Interfaz*	
Descripción:	Usado en Fedora para asignar los valores de las variables de red específicas para cada interfaz de red (recogida de valores de red mediante DHCP, BOOTP o local), dirección IP, máscara de red, dirección de difusión, etc.
Formato:	*Variable*=*Valor* ...
/etc/network/interfaces	
Descripción:	Fichero equivalente al anterior, usado en Ubuntu para configurar todas las interfaces de red del sistema.
Formato:	[**auto** *Interfaz*] **iface** *Interfaz Parámetros* [*Variable Valor*] ... ...
/etc/hosts	
Descripción:	Almacena la asociación entre dirección IP, nombre y alias de ordenadores conocidos. Siempre debe estar presente la dirección 127.0.0.1.
Formato:	*DirecciónIP Nombre* [*Alias ...*] ...
/etc/resolv.conf	
Descripción:	Establece las bases para la resolución de nombres, indicando dominio del ordenador, dirección de los servidores de nombres y otros dominios de interés.
Formato:	**domain** *Dominio* **nameserver** *IPServidorDNS* ... [**search** *DominioBúsqueda ...*]

/etc/nsswitch.conf	
Descripción:	Indica el orden de búsqueda para ficheros de red.
Formato:	*TipoFichero TipoBúsqueda ...* ...
Tipos de búsqueda:	`files`: archivos locales. `nis`: NIS. `nisplus`: NIS+. `ldap`: servicio de directorios. `dns`: servicio de nombres.
/etc/services	
Descripción:	Indica el protocolo y el puerto utilizados por cada servicio de comunicaciones (este fichero no debe modificarse, ya que suele estar bien configurado).
Formato:	*Servicio Puerto/Protocolo [Alias ...]* ...

4.4. Servicios de red.

Los protocolos definidos para controlar cada servicio de comunicaciones utilizan una especie de punto de anclaje a los protocolos TCP o UDP. Este mecanismo es conocido como **puerto**.

Si una aplicación quiere ofrecer un cierto servicio, se engancha ella misma a un puerto y espera las peticiones de los clientes (escuchar en el puerto). Cuando un cliente quiere usar este servicio, el sistema le asigna un puerto libre en su nodo local y se conecta al puerto del servidor en el nodo remoto.

Un puerto del servidor puede ser abierto por diferentes máquinas, pero no existe la posibilidad de ser usado por más de una de ellas al mismo tiempo [6]. Por lo tanto, para atender las peticiones de múltiples clientes, el servidor puede delegarlas a subprocesos que las gestionan individualmente.

4.4.1. Breve descripción de los principales servicios de red.

Para finalizar el capítulo, la siguiente tabla presenta una sencilla descripción de los servicios de red más utilizados en Linux.

dhcp	
Descripción:	Servicio de asignación remota de parámetros de la red, tanto estática como dinámicamente; utiliza el protocolo DHCP, aunque también puede usar BOOTP.
Fichero de configuración:	`/etc/dhcpd.conf`
ldap	
Descripción:	Servicio de acceso a directorios mediante protocolo LDAP. Un directorio es un árbol donde se incluye todo tipo de recursos agrupados lógicamente.
Fichero de configuración:	`/etc/openldap/slapd.conf`
Directorio de esquemas LDAP:	`/etc/openldap/schemes`
httpd, apache2	
Descripción:	Servicio de acceso a la información mediante hipertexto, utilizando los protocolos HTTP.
Fichero de configuración:	`/etc/`*`Servicio`*`/conf/httpd.conf`, `/etc/`*`Servicio`*`/conf.d/*`
squid	
Descripción:	Servicio de acceso a la información mediante hipertexto, utilizando el protocolo HTTP.
Fichero de configuración:	`/etc/squid/squid.conf`
samba	
Descripción:	Servicio que permite compartir recursos (ficheros e impresoras) mediante los protocolos CISS o SMB.
Fichero de configuración:	`/etc/samba/smb.conf`
ssh	
Descripción:	Servicio para la conexión remota y segura al intérprete de mandatos del sistema mediante Secure Shell.
Ficheros de configuración:	`/etc/ssh/sshd_config, /etc/ssh/ssh_config`

subversion	
Descripción:	Servicio para el control y almacenamiento de versiones y revisiones de ficheros, soporta accesos SVN y WebDAV.
Fichero de configuración:	Configuración de acceso WebDAV mediante Apache.

5. Arranque y servicios.

5.1. Proceso de arranque.

Durante el proceso de arranque de la máquina se realizan las comprobaciones necesarias para configurar y activar todos los servicios definidos por el administrador del sistema. Es fundamental conocer este procedimiento y preparar los cambios necesarios según la planificación realizada para dicho ordenador.

El proceso de arranque de un servidor Linux basado en arquitectura x86 (Intel, AMD) comprende los siguientes pasos [2]:

- Tras comprobar los dispositivos de la máquina, el BIOS ejecuta el primer paso del cargador del sistema, situado en el sector de arranque del primer disco duro. GRUB es el cargador usado actualmente en Linux.
- El cargador de arranque ejecuta el segundo paso del proceso, situado en la partición `/boot`.
- El núcleo de Linux y sus módulos adicionales se cargan en memoria y se monta la partición raíz en modo de sólo lectura.
- El núcleo toma el control de la secuencia de arranque y ejecuta el proceso de inicio (**`/sbin/init`**.
- Este programa de iniciación carga todos los servicios definidos, ejecuta los programas de iniciación y configuración y monta las particiones definidas. **`/sbin/init`** lee la información de los ficheros de configuración (`/etc/inittab` en SysVinit de Fedora o `/etc/event.d/*` en Upstart de Ubuntu). El proceso carga en primer lugar los servicios básicos y luego los asociados al nivel de arranque elegido por el administrador.
- Se finaliza el arranque del sistema presentando al usuario el proceso de conexión (**`login`**) o el entorno gráfico (normalmente GNOME o KDE).

Evidentemente, por motivos de seguridad, todos los ficheros y mandatos de configuración tienen que estar completamente vetados para los usuarios normales del servidor y sólo pueden ser ejecutados o modificados por el usuario **`root`**.

SysVinit, implantado por defecto en Fedora y en muchas otras distribuciones Linux, es un proceso de arranque secuencial (basado en System V Init), que es lento y presenta ciertos problemas para usar nuevos dispositivos (como sistemas de archivos en USB).

Ubuntu utiliza **Upstart**, un proceso controlado por eventos que aún está en proceso de desarrollo y que puede emular el funcionamiento de SysVinit.

Existen también otras alternativas en desarrollo basadas métodos de arranque paralelo (**init-NG**) o dirigidas por eventos (**Solaris Service Management Facility** o **Apple launchd**).

5.2. El cargador GRUB.

Como se ha descrito en el apartado anterior, un programa cargador de arranque es el encargado de iniciar un sistema operativo.

Los principales cargadores usados por Linux soportan la definición de un menú de ejecución para los distintos sistemas operativos instalados en el ordenador e iniciarlos en determinados niveles de ejecución.

El **Cargador de Arranque Unificado de GNU** (**GRUB**) [1] permite al usuario elegir el sistema operativo y el núcleo con que desea trabajar.

GRUB tiene 2 modos de operación [2]:

- El **modo directo** se usa para cargar el núcleo de Linux sin ningún tipo de intermediarios.
- El **modo encadenado** se utiliza para cargar otros sistemas operativos y apunta al primer sector de arranque de la partición, donde se encuentran los ficheros de iniciación del sistema.

GRUB ofrece al usuario un entorno de operación válido para realizar configuraciones previas al inicio del sistema operativo, pudiendo acceder directamente a su fichero de control `/boot/grub/grub.conf` o `/boot/grub/menu.lst`.

Las órdenes para la configuración de GRUB definen las características del menú de arranque, indicando los distintas opciones de carga de sistemas, la opción por omisión, límites temporales, entorno gráfico, etc.

El siguiente ejemplo presenta la configuración de un menú para arrancar Ubuntu 10.04 Lucid con un núcleo versión 2.6.32-22 en la segunda partición del primer disco, y Windows XP en la primera partición.

```
boot=/dev/sda
default=0
timeout=5
title Ubuntu 10.04 LTS, kernel 2.6.32-22-generic
  kernel /boot/vmlinuz-2.6.32-22-generic root=/dev/sda2 ro quiet
  initrd /boot/initrd.img-2.6.32-22-generic
  quiet
title Windows XP SP3
  root (hd0,1)
  makeactive
  chainloader +1
```

Cuando se enciende la máquina, GRUB presenta al usuario un menú de selección y éste puede editar las distintas opciones antes de arrancar el sistema operativo correspondiente.

GRUB 2 es una versión en desarrollo, incluida en las nuevas distribuciones de Linux, que supone una redefinición del cargador, para permitir su uso en diferentes arquitecturas, acceder a los sistemas de ficheros locales, incluir un lenguaje de *scripts* complejo, etc.

5.3. El Núcleo.

Los sistemas operativos Unix se basan en una estructura de capas, donde las capas internas prestan servicios básicos a las externas. El **Núcleo** (*kernel*) es la parte principal del sistema operativo, realiza las funciones básicas de control y presta los servicios esenciales para gestionar el sistema operativo.

El núcleo de Linux consta principalmente de los componentes descritos en la siguiente tabla [10].

Gestor de memoria:	Encargado de asignar áreas de memoria y de espacios de paginación a los procesos, a los módulos del núcleo y al área de *caché*.
Gestor de procesos:	Parte esencial que crea, activa y termina los procesos; implementa las reglas de la multitarea.
Controladores de dispositivos:	Gestionan la comunicación del sistema con cada uno de los dispositivos conectados. El núcleo se configura en el proceso de arranque para cargar los módulos necesarios para controlar los dispositivos específicos de cada máquina.
Gestor del sistema de archivos virtual:	Capa intermedia que permite acceder uniformemente a los sistemas de archivos, manteniendo un árbol de directorios homogéneo.
Gestor de redes:	Capa abstracta para el acceso general a la red informática, independientemente del tipo de dispositivos usado y de la arquitectura de la red.

Siendo Linux un sistema operativo basado en el código abierto, el administrador puede rehacer el núcleo, incluyendo o eliminando características operativas, según sus necesidades. El proceso para recompilar el núcleo de Linux es cada de vez más sencillo de realizar, ya que se configura mediante menús con una gran cantidad de opciones.

Para aumentar las prestaciones del sistema, es conveniente contar con un núcleo pequeño, pero que pueda tratar todas las funciones básicas del sistema. La gestión de los dispositivos y de las funciones adicionales puede ser controlada por los módulos del núcleo.

5.3.1. Módulos.

Los módulos del núcleo de Linux son objetos compilados en lenguaje C que controlan elementos o funciones específicas.

Los módulos básicos para el control de los dispositivos conectados se cargan en el proceso de arranque del sistema operativo. El resto de módulos que el gestor del sistema considera necesarios deben enumerarse en los ficheros de configuración, así como los parámetros opcionales que éstos requieran para obtener un funcionamiento óptimo del servidor.

La versión 2.4 del núcleo de Linux usa un único fichero `/etc/modules.conf`, mientras que las distribuciones basadas en la versión 2.6 del núcleo localizan varios ficheros (uno por módulo o grupo de módulos) en el directorio `/etc/modprobe.d`.

En los ficheros de configuración se especifican las asociaciones (alias) entre módulos reales y virtuales. Los módulos virtuales son las interfaces entre el sistema y los módulos que controlan dispositivos reales; son los encargados, por ejemplo, de gestionar la red, los sistemas de archivos especiales, las capacidades exclusivas de tarjetas de sonido, etc.

La siguiente tabla describe las órdenes del sistema implicadas en la gestión de los módulos del núcleo.

Mandato	Descripción
`lsmod`	Lista los módulos cargados.
`modprobe`	Prueba un determinado módulo del sistema e indica si puede ser instalado.
`insmod`	Instala un nuevo módulo.
`rmmod`	Desinstala un módulo cargado.

5.3.2. Parámetros de operación.

Las nuevas versiones de los núcleos de Linux soportan la configuración y personalización de sus parámetros de operación, que afectan a la conducta de sus componentes, tales como: incrementar el máximo

número de ficheros abiertos (`fs.file-max`) o activar la capacidad de reenviar paquetes para crear cortafuegos (`net.ipv4.ip_forward`).

Las modificaciones en el entorno predefinido para el núcleo se verán reflejadas en el fichero correspondiente a la característica modificada, situado en el sistema de archivos virtual `/proc/sys`.

El fichero de configuración de los parámetros operativos del núcleo es `/etc/sysctl.conf` y puede ser editado por el responsable del sistema para incluir los cambios en el próximo arranque del sistema. Asimismo, los cambios se activan automáticamente ejecutando la orden **`sysctl -p`**.

La siguiente tabla muestra el formato de las líneas del fichero `/etc/sysctl.conf`.

Formato	Descripción
`/etc/sysctl.conf`	
Componente[.*Subcomp*].*Parámetro* = *Valor* ...	Configuración de los parámetros del núcleo. Los campos son: 1. Componente principal del núcleo: núcleo (`kernel`), memoria virtual (`vm`), sistema de archivos (`fs`) o red (`net`). Algunos de éstos pueden contar con grupos (directorios) de parámetros. 2. Parámetro de operación. 3. Valor asignado al parámetro.

Algunos programas pueden almacenar también ficheros de configuración de parámetros del núcleo en el directorio `/etc/sysctl.d`, los cuales utilizar una sintaxis similar a la del fichero principal.

Puede obtenerse una completa información sobre los parámetros de operación del Núcleo de Linux en la documentación del código fuente del Kernel y en su página de descarga [x].

5.4. Niveles de arranque en SysVinit.

Muchas distribuciones Linux, como Fedora, utilizar un proceso de arranque secuencial heredado de Unix System V, conocido como **SysVinit**.

Los niveles de arranque sirven para que SysVinit pueda operar de distintas maneras según las necesidades del administrador. Simplemente

cambiando el nivel de arranque, el sistema operativo puede entrar en modo mantenimiento y posteriormente volver al modo multiusuario.

La siguiente tabla describe los niveles de ejecución soportados por SysVInit.

Nivel	Descripción
0	Parada del sistema.
1 o **S**	Nivel de mantenimiento para usuario privilegiado (monousuario).
2	Definido por el administrador en Fedora, por defecto en Ubuntu, multiusuario sin NFS en Solaris y AIX.
3	Nivel de multiusuario, definido por el administrador en Ubuntu.
4	Definido por el administrador o no usado.
5	Nivel de multiusuario con entorno gráfico, definido en Ubuntu.
6	Rearranque del sistema.

El nivel de arranque usado por defecto en el sistema se define en el fichero de configuración `/etc/inittab`. Sin embargo, la orden **`init`** permite modificar el nivel de ejecución de la máquina en cualquier momento.

El mandato **`init`** lee los guiones de configuración almacenados en el subdirectorio `/etc/rc`*`N`*`.d`, correspondiente al nivel de ejecución seleccionado. El modo de operación es el siguiente:

- Parar en secuencia los procesos correspondientes a los ficheros **`K`***`NN`*`servicio`, siendo *`NN`* el orden de la secuencia (2 dígitos) para dicho servicio (se ejecuta: **`/etc/rc`***`N`***`.d/K`***`MM`***`servicio stop`**).
- Arrancar en orden secuencial los procesos de los ficheros **`S`***`NN`*`servicio` (se ejecuta: **`/etc/rc`***`N`***`.d/S`***`MM`***`servicio start`**).

Los guiones de ejecución de servicios se encuentran normalmente en el directorio `/etc/init.d` y están enlazados simbólicamente a los *scripts* de cada nivel de ejecución. De esta manera, el administrador puede arrancar o parar servicios independientemente, ejecutando el *script* correspondiente o el orden **`service`** de Fedora:

```
service Servicio { start|stop|restart|reload|status }
```

La siguiente tabla describe algunos de los programas incluidos en Fedora 13 para la gestión de servicios ejecutados en el proceso de arranque del servidor.

Mandato	Descripción
chkconfig	Establece los enlaces simbólicos para incluir un servicio en los niveles de arranque indicados.
ntsysv	Menú para establecer los servicios que se ejecutarán en los niveles de multiusuario.
service	Arranca o para un determinado servicio.
system-config-services	Interfaz gráfica para la gestión de servicios, incluidos los dependientes de Inetd.
telinit	Cambia al nivel de ejecución especificado.

El siguiente gráfico muestra un ejemplo de arranque de una máquina con Fedora 13.

5.5. Trabajos en Upstart.

Upstart [viii] es un proceso de arranque no secuencial basado en eventos que ha sido incluido por defecto a partir de la versión 6.10 de Ubuntu.

Un trabajo es una tarea o un servicio que puede ser ejecutado o detenido cuando se dispara un evento, y que también puede generar nuevos eventos para gestionar otros trabajos que dependen de él.

Un sistema de arranque de este tipo tiene la ventaja de poder lanzar las tareas necesarias cuando el sistema reconoce los dispositivos correspondientes, sin necesidad de esperar a terminar la ejecución de tareas anteriores. Por lo tanto, puede definirse una especie de árbol de ejecución de tareas.

Upstart puede emular el modo de trabajar de SysVinit, definiendo niveles de ejecución para los servicios generales del sistema. Ubuntu 8.10 Intrepid

define por defecto el nivel 2, además de los niveles de mantenimiento (0, 1 y 6).

El siguiente ejemplo muestra el fichero `/etc/event.d/rc2`, que configura el trabajo para definir el nivel de ejecución 2.

```
start on runlevel 2
stop on runlevel [!2]
console output
script
    set $(runlevel --set 2 || true)
    exec /etc/init.d/rc 2
endscript
```

El administrador puede definir el orden de arranque de los servicios, de manera similar a SysVinit, incluyendo los enlaces simbólicos a los *scripts* correspondientes en el directorio `/etc/init.d/rc`*N*`.d` o ejecutando la orden **`update-rc.d`**.

El formato de ejecución para arrancar o parar un servicio es el siguiente:

```
/etc/init.d/Servicio { start|stop|restart|reload|status }
```

La tabla describe algunos de los mandatos de gestión de servicios y trabajos de arranque en Ubuntu 7.10.

Mandato	Descripción
`initctl`	Utilidad de control del proceso de arranque para ejecutar órdenes y lanzar eventos.
`start`	Arranca un trabajo
`status`	Muestra el estado de ejecución de un trabajo.
`stop`	Para un trabajo.
`telinit`	Cambia al nivel de ejecución especificado.
`update-rc.d`	Gestiona los enlaces para los *scripts* del tipo SysVinit.

5.6. Servicios.

Los servicios son programas cargados en un determinado nivel de ejecución, que suministran al usuario ciertas utilidades o beneficios. Cada servicio supone una posibilidad de conexión con la máquina, lo que también implica la posibilidad de sufrir ataques contra la seguridad del sistema.

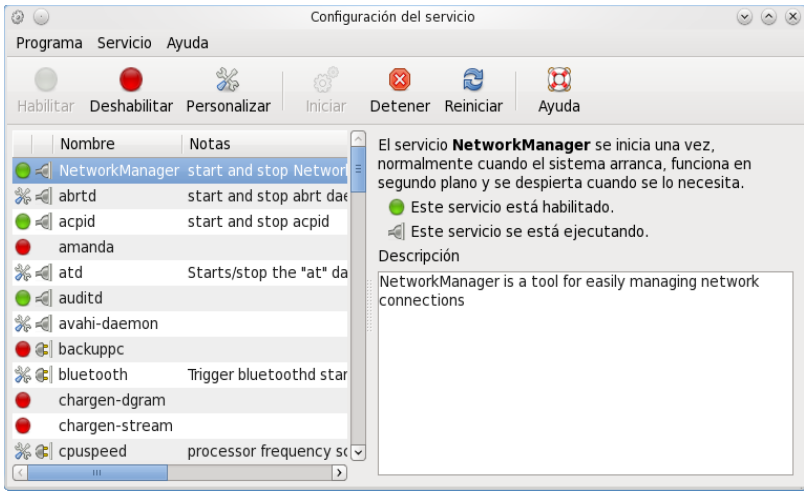
El administrador sólo debe activar los servicios estrictamente necesarios para su máquina.

La siguiente tabla describe de forma breve los servicios más usados en Linux.

Servicio	Descripción
apmd	Control de ahorro de energía.
atd	Planificador de tareas.
avahi-daemon	Cliente para descubrir servicios de configuración de red Zeroconf.
bluetooth	Control de conexiones Bluetooth.
crond	Ejecución cronológica de programas.
cups, cupsys	Servidor de impresión mediante protocolo IPP.
dhcp	Servicio DHCP para la asignación remota de parámetros de la red.
httpd, apache2	Servidor Apache para suministrar acceso a páginas *web*.
inn	Servicio de noticias.
ldap	Servicio de directorios LDAP.
mailman	Servidor de lista de distribución de correo con interfaz *web*.
named	Servidor de nombres de dominio (DNS).
NetworkManager	Servico que mantiene activas las conexiones de red.
nfs	Acceso remoto a directorios mediante NFS.
ntpd	Servidor de sincronización horaria.
postfix	Servidor de correo electrónico.
rsync	Sincronización remota de contenido entre servidores.
sendmail	Servidor de correo electrónico.
sockd	Servidor representante (*proxy*) para aplicaciones.
samba, smb	Servicio para compartir ficheros y recursos, compatible con la red de Windows.

squid	Servidor representante (*proxy*) para accesos mediante HTTP y FTP.
ssh	Conexión segura.
svnserve	Servicio independiente de Subversion (sin Apache).
syslogd	Registro de anotaciones e incidencias.
sysstat	Recopilador de estadísticas de actividad del sistema.
wu-ftpd	Servicio FTP para transferencia de ficheros.
xinetd	Metaservicio de red.
ypserv	Servicio principal para NIS o NIS+.

El siguiente gráfico muestra la aplicación para gestión de servicios **system-config-services** incluida en Fedora 13.



5.7. Control básico de procesos.

Puede entenderse por **proceso** todo programa o mandato en ejecución. Un proceso tiene las siguientes características:

- Cada proceso consta de zona de código, de datos y de pila.
- Los procesos existen en una jerarquía de árbol (varios hijos, un sólo padre).
- El sistema asigna un identificador de proceso (**PID**) único al iniciar el proceso.
- El planificador de tareas asigna un tiempo compartido para el proceso según su prioridad (sólo **root** puede aumentar la prioridad de un proceso).

- Cada proceso almacena su identificador (**PID**) el de su proceso padre (**PPID**), el propietario y grupo del proceso y las variables de entorno.

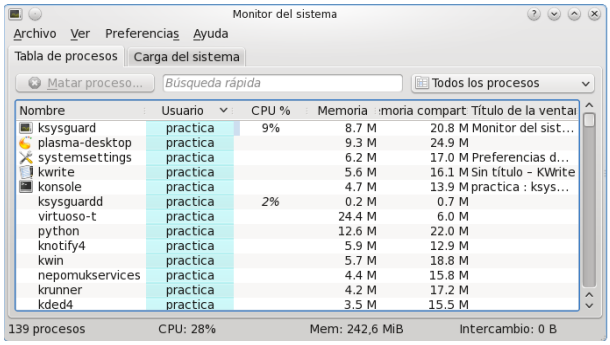
El *superusuario* debe mantener el control del sistema en todo momento, realizando revisiones periódicas de los procesos que se están ejecutando en el servidor, lo que puede evitar problemas y abusos que afecten el funcionamiento normal de la máquina.

La siguiente tabla describe los mandatos más usados para el control de procesos.

Mandato	Descripción
ps	Presenta una lista con los procesos activos en la máquina, indicando propietario, identificador del proceso (PID), identificador del proceso padre (PPID), mandato, etc.
kill	Manda una señal de interrupción a uno o a varios procesos. Suele usarse para finalizar su ejecución.
pgrep	Lista procesos que cumplan un cierto criterio.
pkill	Manda una señal a procesos que cumplan un cierto criterio (tener especial cuidado al ejecutar esta orden).
nice	Cambia la prioridad de los procesos. Suele usarse para bajar el tiempo de ejecución de procesos que saturan al sistema.
top	Presenta una lista actualizada de los procesos que consumen más recursos. También permite mandar señales y modificar la prioridad de ejecución.
lsof	Lista los ficheros y las conexiones de red abiertos por cada proceso, indicando además el propietario, PID, prioridad, mandato, etc.

El mandato **lsof** es una potente herramienta administrativa, ya que muestra todos los ficheros, tuberías con nombre, dispositivos y conexiones de red abiertos por cada proceso. Suele usarse para comprobar aquellos procesos sospechosos de crear problemas y para revisar las conexiones de red de cada servicio.

Por último, el siguiente gráfico muestra la ejecución de **KSysGuard** del entorno KDE 4.4 bajo Fedora 13, que puede usarse para realizar una gestión básica de los procesos del sistema.



6. Referencias.

1. Red Hat Inc.: *"Red Hat Linux 9: The Red Hat Linux System Administration Primer"*, 2.003.
2. Red Hat Inc.: "*Red Hat Linux 9: The Red Hat Linux Reference Guide*", 2.003.
3. R. M. Gómez Labrador: *"Seminario 06013. Programación Avanzada en Shell (Parte I: línea de comandos)", 3ª edición*. Servicio de Formación y Desarrollo del PAS (Universidad de Sevilla), 2.006.
4. R. M. Gómez Labrador: *"Curso 03-12. Administración de Sistemas Linux Red Hat"*. Secretariado de Formación Permanente del PAS (Universidad de Sevilla), 2.003.
5. G. Mourani: *"Securing and Optimizing Linux: The Ultimate Solution, v2.0"*. Open Network Architecture Inc., 2.001.
6. D. Barreña Molina y otros: *"Proyecto RHODAS: Migración a estaciones de trabajo Linux para usuario final en el MAP"*. Ministerio de Administraciones Públicas (España), 2.002.
7. R. Russell, D. Quinlan, C. Yeoh: *"Filesystem Hierarchy Standard, v2.3"*. 2.004.
8. L. Virzenius, J. Oja, S. Stafford: *"The Linux System Administration Guide, v0.7"*. 2.001.
9. SNIA: *"CIFS Technical Reference, v1.0"*, 2.002.

i. Servicio de Formación y Desarrollo del P.A.S. de la Universidad de Sevilla: http://www.forpas.us.es/
ii. Linux OnLine!: http://www.linux.org/
iii. The Linux Documentation Project (TLDP): http://www.tldp.org/
iv. Proyecto HispaLinux (LDP-ES): http://www.hispalinux.es/
v. Filesystem Hierarchy Standard: http://www.pathname.com/fhs/
vi. Proyecto GNU: http://www.gnu.org/
vii. Open Source Initiative: http://www.opensource.org/
viii. Upstart - event-based init daemon: http://upstart.ubuntu.com/
ix. Migrating to Ext4: http://www.ibm.com/developerworks/linux/library/l-ext4/
x. The Linux Kernel Archives: http://www.kernel.org/
xi. The Internet Engineering Task Force: http://www.ietf.org/