

LINUX/LLIUUREX:

REPASO DE CONTENIDOS Y CONCEPTOS BÁSICOS

Jose Luis Gil Gil
IES "Mare Nostrum"
2011



CONTENIDOS

CONCEPTOS BÁSICOS.....	3
SISTEMAS DE FICHEROS	3
ESTRUCTURA DE DIRECTORIOS	4
DISPOSITIVOS EN /DEV	5
USUARIOS EN LINUX	6
COMANDO 'SUDO'	7
PERMISOS EN LINUX	8
COMANDOS EN LINUX.....	10
ALGUNAS OPCIONES COMUNES.....	10
COMANDOS BÁSICOS	10
MANEJO BÁSICO DE FICHEROS Y DIRECTORIOS	11
GESTIÓN DE USUARIOS, GRUPOS Y PERMISOS	11
<i>Comandos de gestión de usuarios.....</i>	<i>11</i>
<i>Comandos de gestión de grupos</i>	<i>12</i>
<i>Comandos de gestión de permisos</i>	<i>13</i>
MONTAJE DE DISPOSITIVOS DE ALMACENAMIENTO	14
INSTALACIÓN DE PAQUETES Y SERVICIOS	16
COMANDOS DE CONFIGURACIÓN DE RED	16
<i>Comandos básicos</i>	<i>16</i>
<i>Configuración de parámetros básicos de red</i>	<i>17</i>
<i>Otros comandos y características de red.....</i>	<i>18</i>

CONCEPTOS BÁSICOS

Sistemas de ficheros

Los sistemas de archivos indican el modo en que se gestionan los archivos dentro de las particiones. Según su complejidad tienen características como previsión de apagones, posibilidad de recuperar datos, indexación para búsquedas rápidas, reducción de la fragmentación para agilizar la lectura de los datos, etc.

Hay varios tipos, normalmente ligados a sistemas operativos concretos. A continuación se listan los más representativos:

fat32 o vfat: Es el sistema de archivos tradicional de MS-DOS y las primeras versiones de Windows. Por esta razón, es considerado como un sistema universal, aunque padece de una gran fragmentación y es un poco inestable.

ntfs: Es el nuevo sistema de Windows, usado a partir del 2000 y el XP. Es muy estable. El problema es que es privativo, con lo cual otros sistemas operativos no pueden acceder a él de manera transparente. Desde Linux sólo se recomienda la lectura, siendo la escritura en estas particiones un poco arriesgada.

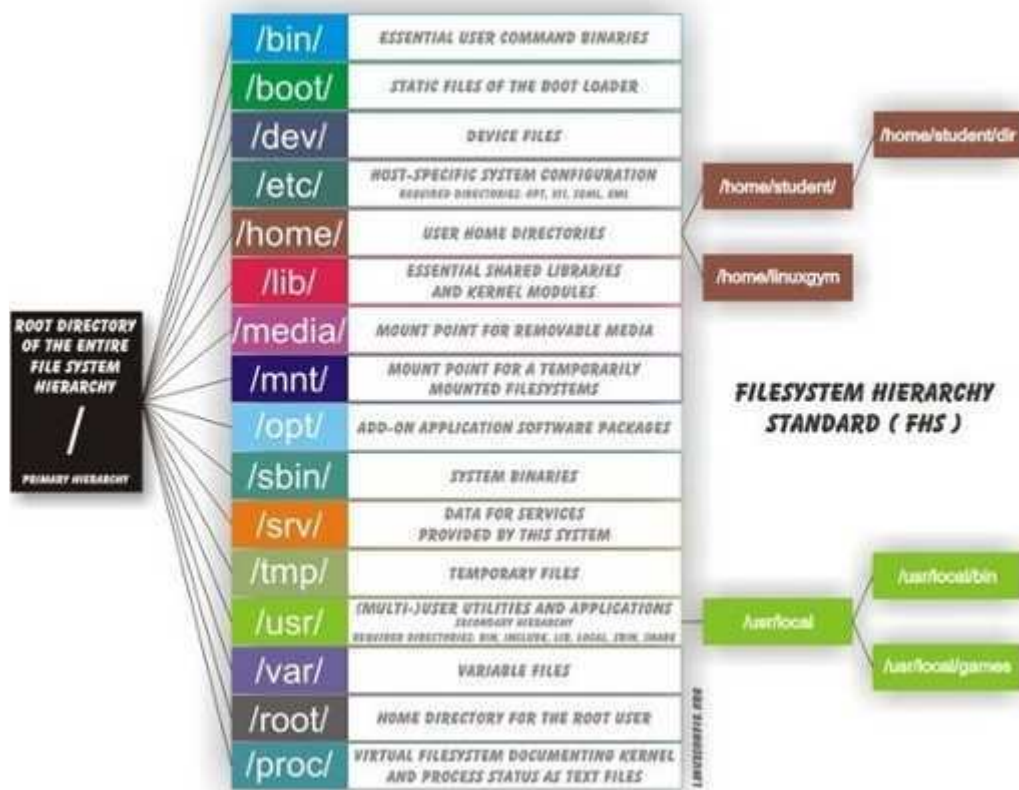
ext2: Hasta hace poco era el sistema estándar de Linux. Tiene una fragmentación bajísima, aunque es un poco lento manejando archivos de gran tamaño.

ext3: Es la versión mejorada de ext2, con previsión de pérdida de datos por fallos del disco o apagones. En contraprestación, es totalmente imposible recuperar datos borrados. Es compatible con el sistema de archivos ext2. Actualmente es el más difundido dentro de la comunidad GNU/Linux y considerado el estándar de facto.

ReiserFS: Es el sistema de archivos de última generación para Linux. Organiza los archivos de tal modo que se agilizan mucho las operaciones con éstos. El problema de ser tan actual es que muchas herramientas (por ejemplo, para recuperar datos) no lo soportan.

swap: Es el sistema de archivos para la partición de intercambio de Linux. Todos los sistemas Linux necesitan una partición de este tipo para cargar los programas y no saturar la memoria RAM cuando se excede su capacidad. En Windows, esto se hace con el archivo pagefile.sys en la misma partición de trabajo, con los problemas que conlleva.

Estructura de directorios



/bin/: contiene programas básicos del sistema ejecutables

/boot/: contiene los ficheros necesarios para el arranque del sistema.

/dev/: contiene los ficheros de dispositivos reconocidos por el sistema (disco duro, lectores cd/dvd, sonido, video...).

/etc/: contiene ficheros y directorios de configuración específicos de nuestro sistema

/home/: directorios personales de los usuarios. Encontraremos una carpeta por cada usuario que haya en el sistema operativo.

/lib/: aquí se encuentran las librerías compartidas y los módulos del kernel.

/media/: en este directorio se montan los dispositivos como el CD-ROM, dispositivos USB, etc.

/opt/: directorio reservado para instalar aplicaciones.

/sbin/: contiene los ficheros binarios ejecutables del sistema operativo.

/srv/: contiene datos específicos que son servidos por el sistema servidor.

/tmp/: directorio de archivos temporales.

/usr/: en este directorio se encuentran la mayoría de los archivos del sistema, aplicaciones, librerías, manuales, juegos... Es un espacio compartido por todos los usuarios del sistema.

/var/: contiene archivos administrativos y datos que cambian frecuentemente (logs, bases de datos, colas de impresión...)

/root/: directorio de root (administrador del sistema).

/proc/: directorio especial donde se almacenan datos del kernel e información sobre procesos.

Algunos archivos importantes en Linux (según versiones) son:

/etc/apt/sources.list: Este archivo configura la lista de repositorios para Añadir aplicaciones mediante aptitude y apt.

/etc/X11/xorg.conf: Este archivo define y configura el entorno gráfico a cargar.

/etc/fstab: Este archivo configura el acceso a los diferentes sistemas de archivos que hay en nuestro sistema.

/etc/passwd y /etc/group: Este archivo controla el uso de usuarios y grupos, contraseñas, con permisos y grupos que pertenecen a cada usuario, archivo muy importante si uno quiere tener un usuario root más que el ya conocido root.

/etc/readahead/boot y /etc/readahead/desktop: Estos archivos contienen la lista de rutas de todos los archivos que se van a cargar en la memoria caché durante el inicio del sistema

Dispositivos en /dev

A diferencia del mundo de Windows lo primero que te sorprenderá es que no está la unidad C:. La distribución de los discos y particiones en Linux es diferente. Ya en la instalación te puedes dar cuenta cómo se nombran los discos y dispositivos en Linux:

Disqueteras

- Primera disquetera: /dev/fd0 (en Windows sería la disquetera A:)
- Segunda disquetera: /dev/fd1

Discos duros (en general: /dev/hdx#, donde x es el disco y # es la partición)

- Primer disco duro: (todo el disco) /dev/hda

Particiones primarias

Primera partición primaria: /dev/hda1

Segunda partición primaria: /dev/hda2

Tercera partición primaria: /dev/hda3

Cuarta partición primaria: /dev/hda4

Particiones lógicas

Primera partición lógica: /dev/hda5

Sucesivamente: /dev/hda#

- Segundo disco duro: (todo el disco) /dev/hdb

Igual que el anterior, /dev/hb1, /dev/hb2...

Puertos y tarjetas

- /dev/eth0 Primera tarjeta de red ethernet
- /dev/eth1 Segunda tarjeta de red ethernet
- /dev/wlan0 Primera tarjeta de red wifi
- /dev/tty0 Puerto COM 1
- /dev/tty1 Puerto COM 2
- /dev/input/mice Ratón USB
- /dev/input/psaux Ratón PS/2

Usuarios en Linux

Hay tres tipos:

Usuario Normal, es un individuo particular que puede entrar en el sistema, con más o menos privilegios que harán uso de los recursos del sistema. *Ejemplo: lliurex, sergio, mrodriguez, etc.*

Usuarios de Sistema, son usuarios propios del sistema vinculados a las tareas que debe realizar el sistema operativo, este tipo de usuario no puede ingresar al sistema con un login normal. *Ejemplo: mail, ftp, bin, sys, proxy, etc.*

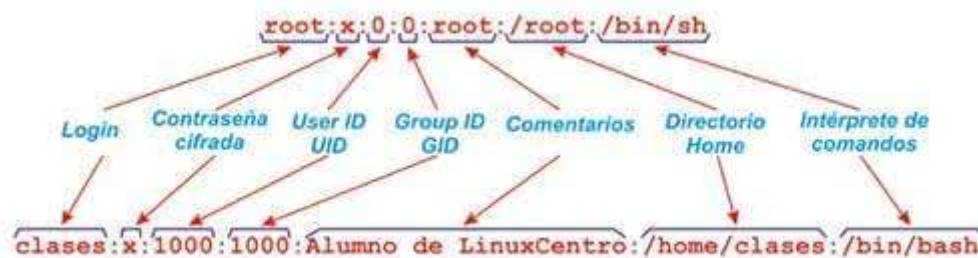
root (superusuario), todo sistema operativo GNU/Linux cuenta con un superusuario, que tiene los máximos privilegios que le permitirán efectuar cualquier operación sobre el sistema, su existencia es imprescindible ya que se encarga de gestionar los servidores, grupos, etc.

Este usuario denominado 'root' que es el administrador absoluto del sistema y que puede hacer y deshacer todo, hasta el punto de ponerlo en peligro.

Hay un fichero donde se pueden ver los usuarios del sistema, “*/etc/passwd*”, y para ver su contenido, sólo hay que poner el comando “*cat /etc/passwd*” y se ve su contenido:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
lliurex:1000:1000:lliurex:/home/clases:/bin/bash
```

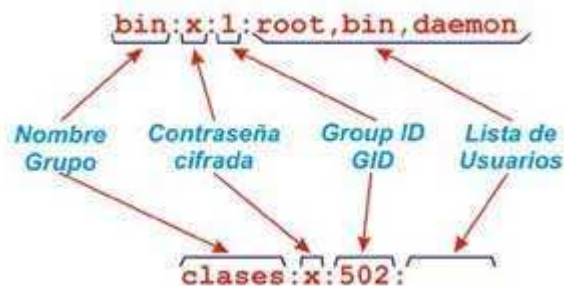
Cada línea esta separada en campos, el separador de campo son los dos puntos (:), y cada campo representa lo siguiente:



Lo mismo sucede en grupos de Linux, con el archivo “*/etc/group*”:

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
clases:x:502:
```

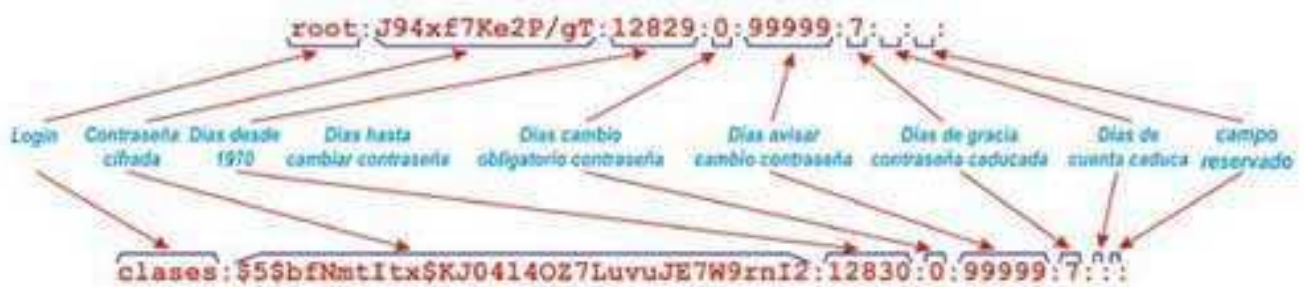
Al igual que el archivo anterior cada línea esta separada en campos el separador de campo son los dos puntos (:), y cada campo representa lo siguiente:



Finalmente, hay un archivo llamado “*/etc/shadow*”, donde, según los permisos que tengamos podremos ver las contraseñas cifradas y otras informaciones:

```
root:$1$qvZCDFha$8CsNHHB/QDYlx3wDnZWzp/:12829:0:99999:7:::
bin:*:12829:0:99999:7::: daemon:*:12829:0:99999:7:::
clases:$1$8Ne4Ij4r$th9obKXkR7iTZGj26jGUc/:12831:0:99999:7:::
```

Igual que los archivos anteriores cada línea esta separada en campos el separador de campo son los dos puntos (:), y cada campo representa lo siguiente



Comando ‘sudo’

El programa **sudo** (abreviatura del inglés *superuser do* o *substitute user do*) es una utilidad de los sistemas operativos tipo Unix, como GNU/Linux, BSD, o Mac OS X, que permite a los usuarios ejecutar programas con los privilegios de seguridad de otro usuario (normalmente el usuario root) de manera segura. Se instala en /usr/bin.

Con ello, si nuestro usuario incluido en el archivo */etc/sudores* tendrá permisos para ejecutar comandos de root de forma segura, impidiendo algunas acciones “suicidas”. Así por ejemplo, si se quiere reiniciar la red, este comando fallaría:

```
/etc/init.d/networking restart
```

pero este otro tendría éxito:

```
sudo /etc/init.d/networking restart
```


Permisos en Linux

Cada archivo o carpeta en Linux tiene sólo tres permisos:

r: (lectura): Cuando el permiso de lectura está activo sobre un directorio significa que se podrá listar los recursos almacenados en él, si está asignado a un archivo se podrá leer su contenido.

w: (escritura): Cuando el permiso de escritura está activo sobre un directorio significa que se podrá crear y borrar archivos en su interior, si esta activado para un archivo significa que se podrá modificar su contenido.

x: (ejecución): Si el permiso de ejecución está activo sobre un directorio significa que el usuario podrá realizar otras funciones dentro de él mediante los otros permisos de lectura y escritura, y si está activo sobre un archivo se podrá ejecutarlo desde la línea de comandos.

Los permisos de los ficheros pueden verse haciendo un “ls -l”, donde se obtiene lo siguiente:

drwxr-xr-x	3	raul	raul	4096	2005-02-16	14:47	Desktop
drwxr-xr-x	5	raul	raul	4096	2005-02-16	14:47	GNUstep
-rw-r--r--	1	raul	raul	246417	2005-03-03	14:47	foto1.png
-rw-r--r--	1	raul	raul	232505	2005-03-03	16:26	carta2.abw
-rw-r--r--	1	raul	raul	239618	2005-03-03	18:15	informe.abw
drwxr-xr-x	2	raul	raul	4096	2005-02-16	14:47	tmp

Diagrama de etiquetado de la salida de `ls -l`:

- Arriba de la primera columna: Tipo de Archivo
- Arriba de la segunda columna: Enlaces
- Arriba de la tercera columna: Propietario
- Arriba de la cuarta columna: Grupo
- Arriba de la quinta columna: Tamaño
- Arriba de la sexta columna: Fecha
- Arriba de la séptima columna: Hora
- Arriba de la octava columna: Nombre
- Debajo de la primera columna: PERMISOS DE PROPIETARIO
- Debajo de la segunda columna: PERMISOS DE GRUPO
- Debajo de la tercera columna: PERMISOS DE OTROS

Por ejemplo, para la línea siguiente en concreto:

```
- rw- r-- r-- 1 raul raul 246417 2008-11-03 13:13 foto1.png
↑ ↑   ↑   ↑   ↑ ↑   ↑   ↑   ↑   ↑   ↑
1 2   3   4   5 6   7   8   9   10  11
```

- 1 : Tipo de archivo = es un archivo regular
- 2 : Permisos propietario = los permisos para el propietario, que son de lectura y escritura
- 3 : Permisos grupo = los permisos de grupo, que tiene permiso de sólo lectura
- 4 : Permisos del resto = el resto de usuarios tienen el permiso de sólo lectura
- 5 : Enlace Físico = tiene un enlace físico
- 6 : Propietario = el usuario raul es el propietario de este archivo
- 7 : Grupo = este archivo pertenece al grupo raul
- 8 : Tamaño = su tamaño es de 246417 bytes
- 9 : Fecha = fue creado el 03 de noviembre de 2008
- 10 : Hora = a 13:13 horas
- 11 : Nombre = el archivo se llama foto1.png

Lo anterior nos deja con 3 preguntas:

¿Quién asigna el propietario y el grupo de cada archivo o carpeta? El propietario de cada archivo o carpeta es el usuario que lo usa, y en el caso de las carpetas o archivos de sistema, suele ser “root” u otro usuario de sistema.

El grupo de cada archivo o carpeta, es igualmente el del usuario que lo crea.

¿Se puede cambiar el propietario o el grupo de un archivo o carpeta? Se puede cambiar el propietario o el grupo de archivos y carpetas, utilizando los comandos **chown** (cambia propietario) y **chgrp** (cambia grupo), que se ven con detalle en la sección de comandos para gestión de usuarios.

¿Cómo se conceden o revocan permisos para usuario, grupo o el resto? Esa acción se realiza mediante el comando **chmod**, en dos modalidades, mediante caracteres (lento) o mediante máscara de número (más rápido). Se ve con detalle en la sección de comandos para gestión de usuarios.

COMANDOS EN LINUX

Algunas opciones comunes

A la hora de ejecutar un comando hay dos opciones importantes:

>, **2>**. Con estos dos caracteres se redirige la salida o los errores de un comando. Ejemplos:

ls -l > listado.txt Este comando no saldrá por pantalla, sino que lo que debería salir por pantalla se guardará en el archivo de texto "listado.txt".

cp doc.txt /home/pepe 2> errores.txt Los posibles errores que diese este comando de copia, no saldrán por pantalla, sino que se almacenarán en "errores.txt".

|. Este carácter, llamado "pipe" o "tubería" sirve para unir dos comandos, de izquierda a derecha, dándole al segundo como entrada lo que sale del primero.

ps -ef | grep gnome Aquí hay dos comandos unidos por "|". El primero "ps -ef" muestra un listado de procesos y el segundo filtra las líneas que contengan la palabra "gnome". Como resultado final, sólo saldrán por pantalla los procesos que contengan la palabra "gnome".

Comandos básicos

ls Sirve para hacer un listado de carpetas y ficheros. Si se pone "-l" salen todos los detalles, de manera similar a MSDOS. Ejemplo

ls -l /etc Muestra todos los archivos y carpetas con detalle en carpeta /etc

clear Limpia la pantalla, al estilo de cls en MS-DOS.

cat <archivo> Visualiza el contenido de un archivo, de manera similar a "type" en MS-DOS.

Con archivos grandes, se usa con "more" (para paginar) o "grep" (para buscar), o bien usar **tail** para ver sólo el final (sobre todo en ficheros de log):

cat archivo.txt Visualiza el contenido del archivo

cat archivo.txt | more Visualiza el contenido del archivo página a página

cat archivo.txt | grep pepe Visualiza sólo las líneas que contengan "pepe"

tail --line 100 /var/log/syslog Muestra las 100 últimas líneas del log de sistema

ps Muestra la lista de procesos que se está ejecutando. Lo mejor, en general es poner como opciones "-ef", para obtener detalle, y filtrar los datos con "more" o "grep", como el comando anterior:

ps -ef Lista de procesos detallada

ps -ef | more Lista de procesos detallada página a página

ps -ef | grep gnome Lista detallada de procesos que contengan "gnome"

nano, joe, emacs, vi Editores de texto, de más fácil de usar a más difícil de usar. Sirven para editar ficheros con mayor o menor complejidad. Todos arrancan igual:

nano archivo.txt Abre “archivo.txt” para edición.

joe archivo.txt Abre “archivo.txt” para edición.

emacs archivo.txt Abre “archivo.txt” para edición.

vi archivo.txt Abre “archivo.txt” para edición.

En caso de querer usar algún editor y no estar disponible, se puede instalar mediante apt o aptitude, como se ve posteriormente en la sección de instalación de paquetes software y servicios.

man Sirve para obtener ayuda detallada sobre comandos de Linux. Cuando acabas de ver la ayuda, se sale con la letra “q”. Por ejemplo:

man ps Nos dice para qué sirve la función “ps” con todas sus opciones.

man chown Nos dice para que sirve la función “chown” con todas sus opciones.

Manejo básico de ficheros y directorios

mkdir <carpeta>, cd <carpeta>, rmdir <carpeta>. Respectivamente, crea, entra o elimina la carpeta especificada. Si no es muy engorroso, es recomendable poner siempre la ruta completa, desde “/”, para evitar confusiones.

pwd Te dice la ruta completa de la carpeta en la que estás actualmente.

cp <origen> <destino>, mv <origen> <destino> Copia o mueve de origen a destino de manera similar al “copy” o “delete” del MS-DOS.

find <carpeta> -name <texto> Busca todos los archivos carpetas que incluyan la palabra “texto” dentro del directorio “carpeta”. Ejemplo:

find /home -name *lliurex* Busca archivos o carpetas que contengan “lliurex” dentro de la carpeta “/home”.

du -csh <carpeta> Indica el tamaño de los archivos que hay en esa carpeta y subcarpetas que contiene.

df -h Indica qué discos y particiones hay y el porcentaje de espacio utilizado

tar xvf <archivo tar> Extrae el contenido del archivo tar en la carpeta actual

tar czf <archivo tar> * Comprime en formato tar.gz el contenido de la carpeta actual y todas las subcarpetas que contiene, conservando permisos, usuarios, etc

grep -rLZ 'texto' <carpeta> Uso de grep para buscar todos los archivos que contienen el texto ‘texto’ dentro de una carpeta y todas sus subcarpetas.

Gestión de usuarios, grupos y permisos

Existe una serie de comandos imprescindibles para la creación de usuarios y otorgamiento de permisos sobre archivos y carpetas.

Comandos de gestión de usuarios

Los más importantes son:

useradd Este comando sirve para dar de alta un usuario, pudiendo especificar, entre muchas otras cosas, su carpeta home y el grupo o grupos al que pertenece. Ejemplo:

useradd pepe Crea el usuario “pepe”

useradd pepe -d /home/pepe -m Crea el usuario “pepe”, y su carpeta home será “/home/pepe”, forzando su creación si no existiera de antes.

useradd pepe -g contables -G administrativos,empleados Crea el usuario “pepe”, indicando que su grupo principal será “contables”, aunque también pertenece a los grupos “administrativos” y “empleados”

passwd Sirve para definir la contraseña de un usuario. Ejemplo:

passwd pepe Sirve para que el sistema nos pregunte una nueva contraseña para el usuario “pepe”.

usermod Sirve para modificar cualquier dato del usuario, excepto su contraseña:

sudo usermod -G jefes pepe Mete a “pepe” en el grupo de “jefes” será

sudo usermod -d /home/otro La nueva carpeta home de “pepe” “/home/otro”

userdel Sirve para borrar un usuario del sistema

sudo userdel pepe Borra al usuario “pepe”.

sudo userdel -r pepe Borra al usuario “pepe” y su carpeta home.

Comandos de gestión de grupos

Los más importantes son:

groupadd Este comando sirve para dar de alta un grupo. Ejemplo:

sudo groupadd jefes Crea el grupo “jefes”

groupdel Sirve para borrar un grupo del sistema

sudo groupdel jefes Borra el grupo “jefes”.

Comandos de gestión de permisos

Como se ha visto, todo archivo o carpeta en Linux, tiene permisos diferenciados para su propietario, para los de su grupo y para el resto de usuarios. Recordemos:

drwxr-xr-x	3	raul	raul	4096	2005-02-16	14:47	Desktop
drwxr-xr-x	5	raul	raul	4096	2005-02-16	14:47	GNUstep
-rw-r--r--	1	raul	raul	246417	2005-03-03	14:47	foto1.png
-rw-r--r--	1	raul	raul	232505	2005-03-03	16:26	carta2.abw
-rw-r--r--	1	raul	raul	239618	2005-03-03	18:15	informe.abw
drwxr-xr-x	2	raul	raul	4096	2005-02-16	14:47	tmp

Diagrama de anotación de la tabla:

- Arriba de la primera columna: Tipo de Archivo
- Arriba de la segunda columna: Enlaces
- Arriba de la tercera columna: Propietario
- Arriba de la cuarta columna: Grupo
- Arriba de la quinta columna: Tamaño
- Arriba de la sexta columna: Fecha
- Arriba de la séptima columna: Hora
- Arriba de la octava columna: Nombre
- Debajo de la primera columna: PERMISOS DE PROPIETARIO
- Debajo de la segunda columna: PERMISOS DE GRUPO
- Debajo de la tercera columna: PERMISOS DE OTROS

En principio, el dueño de un archivo y el grupo de un archivo son el usuario que lo creó y su grupo, pero eso se puede cambiar con dos comandos:

chown Sirve para cambiar el propietario de un archivo. Ejemplo:

sudo chown pepe archivo.txt Hace que “pepe” sea el nuevo dueño de “archivo.txt”

chgrp Sirve para cambiar el grupo de un archivo. Ejemplo:

sudo chgrp jefes archivo.txt Hace que el grupo “jefes” sea el nuevo grupo de “archivo.txt”

Sólo falta saber cómo alterar los permisos en las tres tripletas (dueño, grupo y otros). La única forma de hacerlo es mediante el comando **chmod**, que tiene dos usos, según a lo que estemos acostumbrados:

chmod: uso fácil, pero lento y engorroso. Sólo hay que especificarle qué permiso queremos cambiar: *lectura (r)*, *escritura (w)* o *ejecución (x)* y a quién se lo queremos aplicar: *(u)usuario*, *(g)grupo*, *(o)tros* o *todos a la vez (a)*, especificando “+” para dar permiso, y “-” para quitarlo. Ejemplos:

sudo chmod a-w archivo.txt Quita a todos los usuarios los permisos de lectura.

`sudo chmod u+w archivo.txt` Da permisos de escritura al usuario dueño de “archivo.txt”.

`sudo chmod g+w archivo.txt` Da permisos de escritura al grupo de “archivo.txt”.

`sudo chmod o+x archivo.txt` Da al resto de usuarios, permisos de ejecución sobre “archivo.txt”.

`sudo chmod a+r archivo.txt` Da a todos los usuarios (dueño, grupo y otros) permisos de lectura sobre “archivo.txt”

chmod: uso más complejo, pero muy rápido. Consiste en especificar un número del 0 al 7 para cada tripleta de permisos, conforme a la tabla siguiente:

Valor	Permisos
0	---
1	--x
2	-w-
3	-wx
4	r--
5	r-x
6	rw-
7	rwX

Esta tabla representa *todas las combinaciones posibles de permisos que puede haber en cada tripleta*.

Por ejemplo, los últimos 4 comandos de los ejemplos de chmod implican:

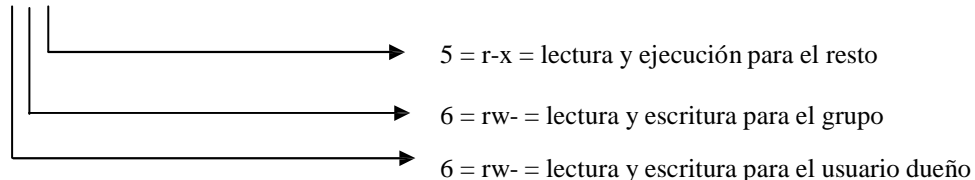
Al usuario dueño, se le conceden lectura y escritura `rw-` **6**

Al grupo, se le concede también lectura y escritura `rw-` **6**

Al resto, se le concede lectura y ejecución `r-x` **5**

Pues bien, los últimos 4 comandos de los ejemplos anteriores, se resumen en 1 sólo:

`chmod 665 archivo.txt`



Montaje de dispositivos de almacenamiento

Lo primero que hay que entender es que Linux, en general, no tiene acceso directo a todos los discos y particiones que haya en el ordenador, ni cdrom o dvd, ni lápices usb.

La filosofía de Linux se basa en acceso únicamente a la partición en la que está instalado, y ***si el usuario quiere tener acceso a más discos, particiones o dispositivos, entonces deberá montarlos él mismo.***

Es una filosofía similar a Windows en el caso de introducir un lápiz USB: es un disco que antes no existía y lo monta para su uso temporal.

Es un proceso mucho más sencillo de lo que parece y tiene 3 simples pasos:

1. **Crear una carpeta donde podamos montar el dispositivo** o partición a la que queremos acceder. En general, por motivos de limpieza y organización, se suele crear dentro de la carpeta “/mnt” o “/media”.

Por ejemplo, si quiero hacer una carpeta para montar mi cdrom, crearé la carpeta siguiente (por ejemplo):

```
sudo mkdir /mnt/micdrom
```

2. **Montar el dispositivo en la carpeta recién creada.** El dispositivo a montar tomará la nomenclatura que se explicó en apartados anteriores. Por ejemplo:

<i>/dev/hda1</i>	Significa primer disco duro, primera partición
<i>/dev/hda2</i>	Significa primer disco duro, segunda partición
<i>/dev/hdb1</i>	Significa segundo disco duro, primera partición
<i>/dev/hdb2</i>	Significa segundo disco duro, segunda partición
<i>/dev/cdrom</i>	Es un atajo rápido que se usa para definir al cd o dvd

Si no se especifica nada, se monta como lectura y escritura, pero puede ser de sólo lectura, en el caso de un cdrom:

```
sudo mount -o ro /dev/cdrom /mnt/micdrom
```

Este comando monta el cdrom en la carpeta /mnt/micdrom, es decir, haciendo un “*ls -l /mnt/cdrom*” nos aparecerán los archivos del cdrom y podremos acceder a ellos, copiarlos, etc.

La opción “-o ro” indica que se monte para sólo lectura (**read only**) y se usa sólo para cdroms y dispositivos similares. Si no la pones, no pasa nada, te avisa y listo.

3. **Desmontar el dispositivo cuando hayamos terminado.** Cuando no queramos usar más el dispositivo, hay que desmontarlo. Es algo similar a lo de “quitar hardware con seguridad” en Windows antes de quitar un lápiz USB.

En este caso, para desmontar el cdrom:

```
sudo umount /mnt/micdrom
```

En caso de que queramos que se monte automáticamente algún dispositivo al iniciar Linux, es necesario editar el archivo */etc/fstab* añadiendo, por ejemplo esta línea:

```
/dev/cdrom /mnt/micdrom cdfs defaults 0 0
```

Instalación de paquetes y servicios

Existen dos herramientas similares para la instalación y desinstalación de paquetes software: *apt* y *aptitude*. Si bien parecen iguales, se suele convenir que *aptitude* es mejor porque tiene en cuenta las dependencias entre paquetes instalados y limpia todas las configuraciones al desinstalarlos.

El primer paso para usar *aptitude* es tener activado todos los sitios web donde queremos que busque los paquetes disponibles. En ubuntu/lliurex, esto se puede hacer de dos formas:

Mediante el fichero de configuración **/etc/apt/sources.list**

Mediante el menú “Sistema”-> “Administración”->”Gestor de paquetes Synaptics”, y dentro del mismo, añadiendo todos los sitios web en la opción de menú “Configuración”->”Repositorios”.

Una vez dados de alta todos los sitios web donde debe buscar paquetes, es posible actualizar la lista de paquetes disponibles poniendo:

```
sudo aptitude update
```

Con todos los paquetes disponibles, podemos buscar alguno de ellos en concreto, de la siguiente manera:

```
sudo aptitude search <nombre>
```

Por ejemplo, *sudo aptitude search telnet* busca todos los paquetes relacionados con el telnet.

Una vez que sepamos qué paquetes queremos instalar sólo hay que dar la orden:

```
sudo aptitude install paquete1 paquete2 ...
```

Si queremos desinstalar un paquete sólo hay que dar la orden (con la opción “purge” elimina toda las configuraciones que dejó ese paquete):

```
sudo aptitude remove paquete --purge
```

Comandos de configuración de red

Comandos básicos

Al igual que en Windows, el comando es “*ipconfig /all*”, en Linux el comando es “*ifconfig*” y nos muestra todas las tarjetas de red, con dirección MAC, IP, IPV6, máscara de subred, etc:

```
ifconfig
```

También como en Windows, cuando se hace un cambio en la red hay que reiniciarla (es el pequeño espacio de tiempo sin red que podemos ver en Windows). **Para reiniciar la red en Linux hay que usar el comando:**

```
sudo /etc/init.d/networking restart
```

Si lo que queremos es **desactivar una tarjeta de red, o volver a activarla**, los comandos adecuados son los siguientes:

```
ifconfig eth0 up      Activa la tarjeta de red “eth0”
```

```
ifconfig eth0 down    Desactiva la tarjeta de red “eth0”
```

Configuración de parámetros básicos de red

A la hora de configurar una red en cualquier sistema operativo necesitamos definir en el equipo la dirección IP, máscara de subred, puerta de enlace y servidor dns. He aquí los comandos necesarios para conseguirlo (después de alguno de ellos puede ser buena idea reiniciar la red):

Especificar la dirección de red y máscara de subred. Se utiliza el propio comando “*ifconfig*”. Un ejemplo que le pone dirección a la tarjeta de red “eth0”:

```
sudo ifconfig eth0 192.168.1.100 netmask 255.255.255.0
```

Si se desea que coja una dirección automática por DHCP, entonces hay que desactivarla y activarla de nuevo, como se ha visto en el apartado anterior.

Especificar la puerta de enlace. Para ello, se utiliza el comando **route**, que usado sin parámetros muestra la tabla de encaminamiento del equipo. Un ejemplo en el que se añade la puerta de enlace es:

```
sudo route add default gw 192.168.1.254
```

Es importante notar que en Linux se pueden añadir tantas puertas de enlace como se quiera, pero si alguna se deja de usar, se puede eliminar así:

```
sudo route del default gw 192.168.1.254
```

Especificar servidor DNS. Para ello, hay que editar el fichero */etc/resolv.conf*, donde se tiene que poner al final una línea “*nameserver*”.

Por ejemplo, para añadir el DNS 80.58.32.97, se añadiría al final de */etc/resolv.conf*, la siguiente línea:

```
nameserver 80.58.32.97
```

Configuración de parámetros básicos de red de forma definitiva en fichero de configuración

Los comandos anteriores permiten definir los parámetros básicos de red de manera volátil, es decir, no se conservarán tras un reinicio de la máquina. Para lograr fijarlos de manera definitiva hay que hacerlo editando el fichero de configuración de red:

```
sudo nano /etc/network/interfaces
```

Allí hay que definir todos los interfaces (tarjetas) de red del sistema, que como hemos visto en el apartado de “dispositivos en /dev”, que denominarán normalmente “eth0”, “eth1”, etc o bien “wlan0”, “wlan1”, etc.

Lo primero que nos encontramos es la configuración del dispositivo de red localhost, llamado “lo”:

```
auto lo
```

Eso hay que dejarlo tal cual, hay que avanzar hacia la configuración del resto de tarjetas de red. Por ejemplo, centrándonos en una hipotética – y habitual - “eth0”, una típica configuración automática bajo DHCP sería:

```
auto eth0  
iface eth0 inet dhcp
```

El funcionamiento es bastante intuitivo simplemente aclarar que “*auto eth0*” especifica simplemente un mapeo automático de la tarjeta de red dentro de los dispositivos, y debe usarse en la mayoría de casos.

Igualmente, resulta sencilla una configuración con IP estática bajo un rango de clase C privado típico:

```
auto eth0  
iface eth0 inet static  
address 192.168.1.5  
netmask 255.255.255.0  
gateway 192.168.1.254
```

Como se puede ver de forma sencilla, simplemente se especifica la dirección IP, máscara de subred y puerta de enlace.

Al terminar de configurar el fichero, será necesario reiniciar la red, con el comando habitual:

```
sudo /etc/init.d/networking restart
```

Otros comandos y características de red

Existen opciones similares a las que estamos acostumbrados a usar en Windows y se detallan a continuación.

-Especificación de nombres de red. En el archivo `/etc/hosts` es posible incluir traducciones de nombre de equipo por dirección ip. Por ejemplo se podrían añadir las siguientes líneas al final de `/etc/hosts`:

```
192.168.1.254  puerta_enlace
80.50.1.23     servidor_dns
```

De esa forma al hacer, por ejemplo, un ping a “puerta_enlace”, lo podría traducir a la dirección 192.168.1.254

-Comando ping. Comprueba la conectividad con equipos remotos. La diferencia consiste en que aquí el ping no manda 4 paquetes, sino que está indefinidamente hasta que le damos a Control-C (o usar -n). Ejemplo:

```
ping www.google.es
ping -n 4 www.google.es (sólo mandará 4)
```

-Comando traceroute. Realiza una traza por todos los saltos intermedios hasta llegar al destino. **Es igual que Windows.** Ejemplo:

```
traceroute www.google.es
```

-Comando route. Muestra la tabla de encaminamiento del equipo, incluida la puerta de enlace para salir a internet. **Es igual que Windows.** Ejemplo:

```
route
route -n
```

-Comando netstat. Muestra las conexiones del equipo con sus respectivos puertos y estadísticas por tarjeta de red (se suele usar con grep para buscar):

```
netstat -i      Estadísticas por tarjeta de red.
netstat -ta     Lista de conexiones y puertos.
netstat -an     Conexiones a/desde el host
```

-Comando lsof. Muestra qué proceso opera en un puerto determinado

```
lsof -n -i:80   Programa que opera en el puerto 80 (protocolo HTTP)
```