

## CONTENIDO

<b>COMANDOS BÁSICOS DE GNU-LINUX.</b>	<b>2</b>
<b>Introducción.</b>	<b>2</b>
<b>Formas de obtener ayuda en linux.</b>	<b>4</b>
<b>Usuarios y grupos.</b>	<b>5</b>
Fichero de usuarios: /etc/passwd	7
El fichero de contraseñas /etc/shadow.	8
El fichero de grupos /etc/group.	9
El fichero de contraseñas de grupos /etc/gshadow.	9
<b>Comandos para administrar usuarios y grupos</b>	<b>10</b>
Comando useradd	10
Comando userdel	10
Comando passwd	10
Comando usermod	11
Comando chfn	11
Comandos groupadd y groupdel	11
Comando groupmod	11
Comando gpasswd	12
Comando su	12
Comando suDO	12
Comando newgrp	13
Comando id	13

## COMANDOS BÁSICOS DE GNU-LINUX.

### INTRODUCCIÓN.

Una vez instalado e inicializado un sistema Linux se dispone de dos vías fundamentales de interacción: una gráfica (si se instaló una interfaz X y se configuró adecuadamente) y una texto conocida como consola o terminal.

Al igual que Unix, Linux ofrece el mecanismo de consolas o terminales virtuales. Este consiste en que a partir de una entrada estándar (el teclado) y con una salida estándar (el monitor) se simulen varias terminales, donde el mismo, o distintos usuarios puedan conectarse indistintamente. De esta forma es posible tener más de una sesión abierta en la misma máquina y trabajar en ella indistintamente. Este mecanismo también facilita la característica multiusuario del sistema Linux pues las diferentes conexiones se pueden establecer con diferentes usuarios.

Por defecto, las consolas desde la uno a la seis tienen asociado un programa que permite conectarse al sistema en modo texto, mientras que la siete, si se instaló y activó el "modo gráfico", constituye una consola gráfica.

El cambio de una consola a otra se realiza a través de la combinación de teclas Alt y Fx (las teclas de Función), donde x oscila entre 1 y 12. De esta forma se pueden acceder un total de 24 consolas virtuales: para las doce primeras se utiliza el Alt izquierdo y para las otras doce el derecho. Por ejemplo para llegar a la consola 16 se presionarían las teclas Alt derecho y F4.

Desde una consola gráfica para cambiar a otra tipo texto se debe además presionar la tecla Ctrl, pues las combinaciones Alt + Fx son capturadas e interpretadas por las aplicaciones gráficas de otra forma.

Así, si pulsamos Ctrl – Alt – F1 accedemos a la consola número 1 (tty1), con Ctrl – Alt – F5 accedemos a la consola número 5 (tty5), etc. Si estamos corriendo Debian en una máquina virtual Vmware nos encontraremos con el problema que dicho software utiliza la combinación Ctrl – Alt como hotkey (atajo de teclado) para liberar el cursor del ratón, por lo que tendremos que acceder a la configuración del Vmware y cambiar este hotkey, a por ejemplo, Ctrl – Alt – Mayus.

Con la tecla Alt izquierda combinada con los cursores (derecho e izquierdo) se puede además, realizar un movimiento circular entre todas aquellas consolas que tengan un proceso asociado (texto, gráfico, etc.). Esta combinación no funcionará una vez que entremos en la consola gráfica.

Si accedemos a una consola en modo texto podremos apreciar que en ella se muestra el nombre de la distribución, la versión de la misma, la versión del kernel y la arquitectura de la máquina. También aparecerá el nombre que se le asignó al sistema en la instalación y la palabra login.

Aquí puede entrarse el nombre de un usuario del sistema. Luego se pedirá la contraseña o password de dicho usuario (tened cuidado ya que al entrar dicho password no se muestra ningún eco en la pantalla). Si ambos son válidos se establecerá la conexión y se mostrará lo que se conoce como prompt del sistema, con forma similar a esta:

```
usuario@pinguino:~$
```

Aquí ha abierto sesión un usuario con nombre “usuario”, en una máquina que se llama “pinguino”, está actualmente en el directorio “~” y sabemos que es un usuario normal y no el root por que el prompt termina con un símbolo “\$” (si fuera root terminaría con un símbolo “#”).

Este entorno de texto donde nos encontramos y que nos permite introducir comandos es conocido comúnmente como Shell (caparazón). Este Shell es capaz de interpretar una gran gama de comandos y sentencias. Constituye a su vez un poderoso lenguaje de programación mediante scripts.

GNU-Linux tiene la filosofía de no obligar al usuario a utilizar un programa determinado para cada acción, sino que siempre da la libertad de elegir el programa que queremos utilizar. Lo mismo ocurre con el Shell que vayamos a utilizar para acceder al sistema. El Shell que más se usa es conocido como bash, aunque existen una gran variedad de ellos, como por ejemplo csh, ksh, etc.

Algunas características que merece la pena conocer de bash son:

- ▶ Auto completar durante la escritura. Al teclear uno o varios caracteres se puede pulsar TAB con el objetivo de que en caso de que pueda completarse de forma unívoca un comando, nombre de fichero o una variable (en dependencia del contexto), complete de forma automática (se escriba el resto de la palabra). Si existieran varias posibilidades para completar la palabra, se oirá un sonido y volviendo a pulsar TAB se mostrarán en pantalla todas las posibilidades existentes. En caso de existir muchas posibilidades (por defecto más de 100) se pregunta si se desea mostrarlas todas o no.
- ▶ Historial de comandos. Esta es una facilidad de muchos otros shells que permite el movimiento a través de los últimos N comandos ejecutados, en la sesión actual o en las anteriores. N por defecto es 1000, pero puede modificarse. Para moverse arriba y abajo se suelen utilizar los cursores, y podemos realizar búsquedas con Control [r]. Podemos indicar que se repita una línea de comando con `^n` y también podemos indicar que se repita una línea de comando que empieza por determinada palabra `^palabra`.
- ▶ Poderosas estructuras de control para realizar scripts. (Procesos por lotes).
- ▶ Definición de funciones y alias para comandos. Las funciones permiten definir subrutinas programadas usando el lenguaje de bash y los alias, asociar nombres a llamados a comandos con ciertas opciones y argumentos de forma más nemotécnica o abreviada.

## FORMAS DE OBTENER AYUDA EN LINUX.

Existen múltiples y variadas formas de obtener ayuda en un sistema Linux. A continuación se describen algunas de ellas:

Muchos comandos poseen una opción para mostrar una ayuda breve acerca de su utilización. Esta opción usualmente consiste en utilizar el parámetro `-h`, `--help` ó `-?` tras el nombre del comando.

```
mkdir --help
```

El comando `help`, que muestra en algunos comandos integrados del bash un manual propio.

```
help alias
```

El comando `info` que muestra información sobre los comandos en una estructura de hipertexto.

```
info mkdir
```

El comando `whatis` que nos da una ayuda rápida sobre comandos.

```
whatis cp
```

El comando `apropos`, que dada una palabra busca los comandos relacionados sobre ella.

```
apropos delete
```

El comando `man` muestra un manual bastante amplio acerca de comandos, formatos de ficheros de configuración, llamadas al sistema, etc. Los manuales están disponibles y pueden instalarse en múltiples idiomas. Estos se dividen internamente en secciones. Un mismo objetivo puede estar representado en varias secciones. De no especificarse ninguna sección a través del primer argumento del comando se tomará la primera donde aparezca.

```
man mkdir
```

## USUARIOS Y GRUPOS.

Como ya se ha afirmado Linux es un sistema multiusuario, lo cual permite que varios usuarios puedan conectarse y trabajar en él de forma simultánea. Las conexiones como ya se ha visto se pueden realizar a través de varias terminales locales o utilizando servicios de red como el Telnet y SSH.

Un usuario se caracteriza por su login el cual debe indicar para conectarse al sistema, y por su password o contraseña. Además puede poseer un conjunto de datos adicionales como el domicilio, teléfono, etc.

El usuario con más privilegios en Linux es **root**. Este es el único con derechos suficientes para crear o eliminar a otros usuarios, además de acceder a todo el sistema de ficheros sin ninguna restricción.

En Linux existen grupos de usuarios que permiten otorgar los mismos privilegios a un conjunto de usuarios. Siempre que se añada un usuario al sistema se creará un grupo con su mismo nombre, llamado grupo primario. Durante la creación o posteriormente, se podrá incorporar el usuario a otros grupos secundarios. Así cuando creamos el usuario Olegario, el sistema creará automáticamente un grupo llamado Olegario que contará como único miembro con el usuario creado.

Tanto los usuarios como los grupos se identifican por el sistema a través de un identificador (ID) numérico. El usuario root siempre tiene el ID cero. Cada usuario cuando se conecta al sistema posee un identificador de usuario asociado (uid) y uno o varios identificadores de grupo (gid).

Al añadir un usuario también se creará un directorio base para el mismo con el nombre de su login. Este directorio se coloca por defecto en el directorio */home/nombredelusuario* excepto para root, cuyo directorio base es */root*.

La información asociada a los usuarios en un sistema Linux se guarda en el fichero **/etc/passwd** y las contraseñas y datos afines en **/etc/shadow**. Por su parte la información de los grupos, sus miembros y passwords están en **/etc/group** y **/etc/gshadow** respectivamente. (Vemos a la derecha un ejemplo del fichero **/etc/passwd**).

Para crear un usuario se usa el comando **useradd**, aunque es mucho mejor usar un script que viene instalado en Linux, que se llama **adduser**. Este script controla como funciona useradd y permite realizar funciones avanzadas, como crear automáticamente el directorio del usuario, configurar su perfil, etc.

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
```

Como ejemplo, vamos a realizar el siguiente ejercicio (abrir la consola de root para realizarlo):

- 1) Creamos dos usuarios, uno con nombre margarita y otro con nombre floripondio. En ambos usamos de contraseña 123 por ejemplo.

```
adduser margarita
adduser floripondio
```

- 2) Creamos un grupo con nombre flores, y añadimos a los dos usuarios anteriores a dicho grupo.

```
addgroup flores
adduser margarita flores
adduser floripondio flores
```

- 3) Visualizamos el fichero /etc/passwd y comprobamos como se han creado dos líneas, una para cada uno de los usuarios que hemos creado.

```
cat /etc/passwd
```

- 4) Visualizamos el fichero /etc/shadow y veremos cómo se han creado también dos líneas, una para cada uno de los usuarios.

```
cat /etc/shadow
```

- 5) Visualizamos el fichero /etc/group y comprobamos como se ha creado una línea para el grupo creado, donde además comprobamos que se han añadido como miembros los usuarios.

```
cat /etc/groups
```

Para comprobar que los usuarios se han creado bien, vamos a realizar lo siguiente:

- A. acceder al 4º terminal (Control – Alt – F4), y hacer un login con el usuario margarita. Una vez abierta sesión, ejecutar el comando `whoami` y el comando `id`. Deberíamos comprobar que efectivamente estamos en el grupo flores. Para cerrar la sesión usamos el comando `logout`.

- B. Acceder al 5º terminal, hacer un login con floripondio y comprobad lo mismo.

(Para volver al terminal grafico desde un terminal de texto, accedemos al 7º terminal).

Comprobaremos como toda la gestión de usuarios y grupos de Linux, en realidad se basa en la modificación de una serie de ficheros de texto que están en el directorio /etc. Esto es algo que como veremos, es común a todo Linux, y siempre que configuremos algo, en el fondo estaremos editando ficheros de texto.

La información que se guarda en cada uno de esos ficheros es la siguiente:

## FICHERO DE USUARIOS: /ETC/PASSWD

Este fichero consta de 7 campos separados por el símbolo dos puntos (:).

Nombre	Contraseña	UID	GID	Información	Dir.Personal	Shell
--------	------------	-----	-----	-------------	--------------	-------

- El campo contraseña indica una X siempre, ya que en los sistemas operativos modernos la contraseña no se almacena en este mismo fichero, ya que sería un fallo de seguridad muy grande. En su lugar la contraseña se almacena en el fichero /etc/shadow.
- UID es el número que se le asigna a cada usuario en el sistema.
- GID es el número del grupo principal al que pertenece este usuario. (El grupo principal suele tener el mismo nombre del usuario).
- Información suelen ser varios campos separados por coma, y se utilizan a título informativo, no tienen ningún significado propio importante.
- Dir.Personal es el directorio donde se almacenará el perfil del usuario (sus documentos y configuraciones). Suele ser habitual que sea /home/*Nombredelusuario*.
- Shell indica el programa que se debe ejecutar para ofrecerle un terminal de acceso al sistema a este usuario, normalmente es el bash ya que es el shell que utilizamos por defecto. Si no queremos que un usuario pueda tener acceso a estos terminales, bastaría con indicarle aquí un nombre falso, como por ejemplo /bin/false y de este modo este usuario no podría interactuar con el sistema aunque realizara correctamente el login.

Vemos aquí un ejemplo de las últimas 5 líneas de un fichero /etc/passwd.

```
usuario@debian740:~$ tail -5 /etc/passwd
saned:x:111:117::/home/saned:/bin/false
Debian-gdm:x:112:118:Gnome Display Manager:/var/lib/gdm3:/bin/false
usuario:x:1000:1000:usuario,,,:/home/usuario:/bin/bash
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
mysql:x:114:121:MySQL Server,,,:/nonexistent:/bin/false
usuario@debian740:~$
```

## EL FICHERO DE CONTRASEÑAS /ETC/SHADOW.

Este fichero consta de 8 campos separados por dos puntos.

Nombre	Contraseña	Ult.cambio	Mínimo	Máximo	Aviso	Inactivo	Caducidad
--------	------------	------------	--------	--------	-------	----------	-----------

- Contraseña es la auténtica contraseña del usuario aunque encriptada. Esta contraseña hoy en día se representa en tres partes distintas, separadas por el símbolo del dólar (\$).
  - Id. Esto nos indica el método de encriptación que fue utilizado. Un valor de 1 indica MD5, un valor de 2 Blowfish, 3 es NT Hash, 5 es SHA-256 y 6 es SHA-512.
  - Salt. Este valor se usa por los algoritmos de encriptación y puede ser de hasta 16 caracteres. Este valor se crea aleatoriamente en cada generación de contraseña.
  - Hash. La contraseña en sí misma. MD5 usa 22 caracteres, SHA-256 usa 43, y SHA-512 usa 86.

Si nos encontramos con un asterisco en este campo significa que la cuenta está bloqueada. Lo mismo se puede conseguir colocando una exclamación (!) al principio de la contraseña. Una cuenta sin contraseña la veremos o bien con todo este campo en blanco, o bien con dos exclamaciones consecutivas (!!).

- Ult.Cambio nos indica el día en que se cambió la contraseña por última vez. (Este número indica los días que han transcurrido desde el 1-1-1970).
- Mínimo. El mínimo número de días que hay que esperar para cambiar la contraseña.
- Máximo. El máximo número de días antes de que la contraseña caduque.
- Aviso. Antes de que la contraseña caduque, avisaremos al usuario al llegar a este número de días.
- Inactivo. Una vez que la contraseña caduque esperaremos este número de días antes de bloquearla definitivamente.
- Caducidad. Es una fecha, en la cual la cuenta automáticamente quedará bloqueada. (Este número indica los días que han transcurrido desde el 1-1-1970).

```
root@debian740:/home/usuario# tail -5 /etc/shadow
saned*:16186:0:99999:7:::
Debian-gdm*:16186:0:99999:7:::
usuario:$6$0aLI6P1D$G8TcjA3i0De0J4sQ6JmGy1nmKJwYlw0.0JNREi4dbv3a48b02EBLNw8Bt2C/
VKuaqwx83aTLuIyhhzm1n2IOG/:16186:0:99999:7:::
hplip*:16372:0:99999:7:::
mysql!:16393:0:99999:7:::
root@debian740:/home/usuario#
```

Dado que estos campos son un poco especiales, es más fácil consultarlos para un usuario concreto con el comando `chage`.

```
chage -l usuario
```



## EL FICHERO DE GRUPOS /ETC/GROUP.

Este fichero consta de 4 campos separados por dos puntos.

Nombre del grupo	Contraseña	GID	Miembros
------------------	------------	-----	----------

- Contraseña. al igual que ocurre con los usuarios, los grupos también pueden tener contraseña, y lo normal es no incluir esta contraseña en este fichero, sino en su propio fichero /etc/gshadow.
- GID. Es el identificador numérico del grupo.
- Miembros. Los nombres de los miembros del grupo separados por coma.

```
root@debian740:/home/usuario# tail -5 /etc/group
usuario:x:1000:
lpadmin:x:119:
ssl-cert:x:120:
mysql:x:121:usuario,margarita
margarita:x:1001:
root@debian740:/home/usuario#
```

## EL FICHERO DE CONTRASEÑAS DE GRUPOS /ETC/GSHADOW.

Este fichero consta de 4 campos separados por dos puntos.

Nombre del grupo	Contraseña	Administradores	Miembros
------------------	------------	-----------------	----------

- Contraseña. Una contraseña encriptada con las mismas características que la existente en el fichero shadow. Si existe una contraseña en un grupo, un usuario que se quiera introducir como miembro del grupo tendrá que saber dicha contraseña ya que el sistema se la pedirá (comando newgrp). Si un grupo no tiene una contraseña, solo el root o los administradores del grupo podrán asignar miembros a dicho grupo.
- Administradores. Una lista de usuarios separados por coma que pueden añadir o eliminar miembros al grupo.
- Miembros. Lista de miembros del grupo separados por coma.

```
root@debian740:/home/usuario# tail -5 /etc/gshadow
usuario:!::
lpadmin:!::
ssl-cert:!::
mysql:!::usuario,margarita
margarita:!::
root@debian740:/home/usuario#
```

## COMANDOS PARA ADMINISTRAR USUARIOS Y GRUPOS

### COMANDO USERADD

El comando `useradd` permite añadir nuevos usuarios al sistema, además de establecer la información por defecto de los nuevos usuarios que se añadan. Se encuentra enlazado simbólicamente por el nombre **adduser**. Ambos nombres se pueden emplear indistintamente.

Sintaxis: `useradd [opciones] [login]`

Ejemplos:

<code>adduser pepe</code>	crea el usuario pepe con las propiedades por defecto
<code>useradd -D</code>	muestra las propiedades por defecto de los nuevos usuarios
<code>adduser -D -b /dir</code>	cambia el directorio base por defecto de los nuevos usuarios (/home) a /dir. El directorio /dir debe existir previamente
<code>adduser pedro amigos</code>	añade a pedro al grupo amigos. Esto es una opción que se le añadió a <code>adduser</code> para poder añadir usuarios a grupos rápidamente.

### COMANDO USERDEL

El comando `userdel` permite eliminar definitivamente un usuario del sistema.

Sintaxis: `userdel [opciones] <login>`

Ejemplo:

<code>userdel -r pepe</code>	elimina al usuario pepe y borra su directorio base. Por defecto el directorio base se mantiene
------------------------------	--

### COMANDO PASSWD

El comando `passwd` permite cambiar el password de un usuario. También puede bloquear, desbloquear y deshabilitar una cuenta. Si se invoca sin argumentos se asume el usuario actual.

Sintaxis: `passwd [opciones] [login]` Ejemplos:

<code>passwd pepe</code>	coloca una contraseña para pepe
<code>passwd -d pepe</code>	deshabilita la cuenta del usuario pepe eliminando su password
<code>passwd -l pepe</code>	bloquea la cuenta del usuario.
<code>passwd -u pepe</code>	desbloquea la cuenta del usuario pepe

## COMANDO USERMOD

El comando usermod se emplea para modificar algunas propiedades de los usuarios como: el login, el directorio base, el shell que se inicia al conectarse, los grupos a los que pertenece, la fecha de expiración de la cuenta, etc. También bloquea y desbloquea una cuenta.

Sintaxis: usermod [opciones] <login>

Ejemplos:

usermod -s /bin/csh pepe	coloca el shell csh para el usuario pepe
usermod -G users,disk pepe	señala como grupos secundarios de pepe a users y disk
usermod -e 2001-10-20 pepe	indica que la cuenta de pepe expirará el 20 de octubre de 2001

## COMANDO CHFN

El comando chfn permite cambiar la información de contacto de un usuario. Esta incluye aspectos como: el nombre completo, la oficina de trabajo y los teléfonos. Se almacena en el fichero de usuarios /etc/passwd.

Sintaxis: chfn [opciones] [login]

Ejemplo:

chfn pepe

## COMANDOS GROUPADD Y GROUPDEL

Los comandos groupadd y groupdel añaden y eliminan un grupo en el sistema respectivamente.

Sintaxis:

groupadd [opciones] <grupo>	groupdel <grupo>	Ejemplos:
groupadd admin		
groupdel admin		

## COMANDO GROUPMOD

El comando groupmod permite modificar en un grupo su identificador y nombre.

Sintaxis: groupmod [opciones] <grupo>

Ejemplo:

groupmod -n usuarios users cambia el nombre del grupo users a usuarios

### COMANDO GPASSWD

El comando `gpasswd` permite administrar los grupos. Se puede utilizar para añadir y eliminar usuarios, señalar un administrador e indicar un password de grupo.

Sintaxis: `gpasswd [opciones] <grupo>`

Ejemplos:

`gpasswd -A pepe admin` señala como administrador del grupo `admin` al usuario `pepe`

`gpasswd admin` cambia el password del grupo `admin`

`gpasswd -a joe admin` añade el usuario `joe` al grupo `admin`

### COMANDO SU

El comando `su` permite ejecutar un shell (u otro comando) cambiando los identificadores del grupo y del usuario actual. Si se le pasa `-` como primer argumento ejecuta el shell como un login shell, o sea se creará un proceso de login tal y como ocurre naturalmente cuando un usuario se conecta al sistema. Si no se especifica el login del usuario se asume `root`.

Sintaxis: `su [opciones] [login]`

Ejemplos:

`su -`

`su pepe` ejecuta un shell haciéndonos pasar por el usuario `pepe`

`su -c "cat /etc/shadow"` ejecuta un comando con los privilegios de `root` en lugar de un shell

### COMANDO SUDO

El programa `sudo` 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.

Los usuarios deben confirmar su identidad al ejecutar `sudo` dando **su propia contraseña** antes de ejecutar el programa requerido. Una vez se ha autenticado el usuario, y si el archivo de configuración `/etc/sudoers` permite dar al usuario acceso al comando requerido, el sistema lo ejecuta y lo registra. En un entorno gráfico, se usa `gksudo`.

El archivo de configuración `/etc/sudoers` especifica qué usuarios pueden ejecutar qué comandos en nombre de qué otros usuarios. Como `sudo` es muy estricto con el formato de este archivo, y cualquier error podría causar problemas serios, existe la utilidad `visudo`; ésta permite al usuario `root` editar el archivo y luego revisar su corrección antes de guardarlo.

### COMANDO NEWGRP

El comando newgrp permite iniciar un nuevo shell cambiando el identificador del grupo (gid) del usuario actual. Sólo podrán hacerlo los usuarios que pertenezcan al grupo o en caso de que este tenga una contraseña, aquellos que la conozcan.

Sintaxis: newgrp [grupo]

Ejemplo:

```
newgrp bin
```

### COMANDO ID

El comando id, imprime dado un usuario, sus identificadores de usuario y de grupo principal (gid y uid) así como del resto de los grupos a los que pertenece. Sin argumentos se asume el usuario actual.

Sintaxis: id [opciones] [login]

Ejemplo:

```
id pepe
```