Gestión de usuarios y grupos en GNU/Linux

1. Introducción

Al ser el sistema operativo Linux multiusuario, puede tener **varios usuarios trabajando en el sistema de forma simultánea**, de manera que cada usuario pueda lanzar sus propios procesos. Los usuarios, a su vez, deben pertenecer a un grupo de usuarios o bien pueden pertenecer a varios grupos. Además, como el sistema opera tivo es multitarea, cada usuario puede tener más de un proceso ejecutándose a la vez, por lo que se hace necesario una gestión eficiente de los procesos del sistema.

2. Usuarios y grupos

Para entrar en el sistema operativo Linux normalmente deberemos identificarnos como usuario. En el proceso de instalación se nos pide que **creemos un usuario**, que será el usuario con el que después entraremos en el sistema.

Los usuarios se identifican por un número de usuario llamado **UID** (**User ID**, **identificador de usuario**). El UID es un número entero distinto a cualquier otro UID de otro usuario, que lo identifica específicamente en el sistema. **Cada usuario debe pertenecer a un grupo** obligatoriamente, llamado **grupo primario o principal**, y además **puede pertenecer a otros** grupos, llamados grupos secundarios.

Un grupo es un conjunto de usuarios. Puede haber grupos que solo tengan a un usuario. Cada grupo se identifica por el **GID** (**Group ID**, **identificador de grupo**), que es también un número entero que lo identifica y los distingue de los demás grupos.

Tipos de usuarios

Entre los usuarios dados de alta en el sistema, podemos distinguir entre tres tipos de usuarios:

- ✔ **Usuario root**. También se le denomina **superusuario** o **administrador**. Es el usuario que **cuenta con todos los privilegios sobre el sistema**. Tiene **acceso total** a todo el sistema, se encarga de la administración del mismo, actualización y mantenimiento. Se identifica por que su UID es el 0.

- ✔ **Usuarios del sistema**. No son usuarios en el sentido físico del término. **Se generan al instalar el sistema o cuando se instala cualquier servicio**. Aunque no tienen todos los privilegios, tienen algunos dependiendo del usuario de que se trate. No son usuarios que entran al sistema e inician una sesión, por lo que no suelen tener un shell de inicio de sesión. El **UID que tienen es mayor a 1 y menor a 1000**, salvo para el usuario **nobody** al que se le asigna **el último UID posible**, el **65534**.

- ✔ **Usuarios normales**. Son los **usuarios que se conectarán al sistema introduciendo su nombre de usuario y su contraseña**. Tienen un **directorio de trabajo dentro** del directorio **/home** que es donde trabajan y dentro del cual tienen todos los privilegios. El **UID** que se les asigna es **a partir del 1000**, aunque es configurable. Al instalar el sistema operativo se crea uno por defecto. Un usuario normal solo tiene acceso a los archivos de su propiedad o a los que se les haya dado permiso.

Si en el sistema solo hay un usuario normal, podemos configurar para que no pida nombre de usuario ni contraseña, sino que se entre directamente como ese usuario.

Ficheros de configuración

Son los **ficheros** que el **sistema lee o modifica** a la hora de **gestionar los usuarios y los grupos**.

/etc/passwd

Este fichero contiene en cada una de sus líneas información sobre un usuario. La información se organiza en campos separados por el carácter “:”. Cada línea del fichero tiene la siguiente estructura:

```
login:x:UID:GID:información:directorio_personal:shell_de_inicio
```

Donde cada campo tiene el siguiente significado:

- ✔ **login**: es **nombre del usuario** con el que nos identificamos para entrar en el sistema operativo, que también se conoce como hacer login.

- ✔ **x**: la **x** significa que hay una **contraseña almacenada** y **encriptada** en el **fichero /etc/shadow**.

- ✔ **UID**: número **entero igual o superior a 0**. El **0 está reservado** al usuario **root**. Los números del 1 al 99 están reservados para usuarios especiales del sistema. Para los usuarios normales se utiliza el rango del 1000 en adelante.

- ✔ **GID**: número **entero igual o superior a 0**. El **0 está reservado** al **grupo de root**. Este número representa el número del **grupo primario** al que pertenece el usuario.

- ✔ **Información**: información sobre el usuario. Se muestra separada por comas. Puede no contener ninguna información, por lo que entonces contendría una serie de comas. A este campo también se le llama GECOS. La información contenida en el campo se puede visualizar con el **comando finger**.

- ✔ **directorio_personal**: ruta absoluta del directorio personal del usuario, normalmente en **/home**.

- ✔ **shell_de_inicio**: ruta absoluta del **shell de entrada por defecto para el usuario**. Puede ser cualquiera, pero el más utilizado es el shell bash. Si es /bin/false o /sbin/nologin, también llamado shell nulo, el usuario podría entrar en el sistema pero no a través de una sesión sino que podría ejecutar ciertos procesos a su nombre como por ejemplo **mail**, **ftp**, **news** o **ssh**.

Mostramos por pantalla el contenido del **fichero /etc/passwd** de **forma paginada**, es decir, pantalla a pantalla. Después, hacemos que se muestre, de cada línea del fichero, el nombre de cada usuario, su UID y su shell de inicio. A continuación, mostramos el número de usuarios que hay dados de alta en el sistema.

```
paco@ubuntupaco:~$ more /etc/passwd
paco@ubuntupaco:~$ cut -d ":" -f 1,3,7 /etc/passwd
paco@ubuntupaco:~$ wc -l /etc/passwd
```

/etc/shadow

En este fichero están las **contraseñas de cada usuario encriptadas**. Este fichero debe tener una especial seguridad, por lo que solo **root** o un **usuario que tenga sus privilegios** puede tener permiso de lectura sobre él. Para las contraseñas en Linux, el usuario que la ha creado, cada vez que entre en el sistema, escribe su contraseña y el sistema comprueba su contraseña con la que tiene almacenada en el fichero shadow.

La función del **algoritmo de encriptación** consiste en convertir una cadena de caracteres en otra, que no tiene que ser del mismo tamaño, de manera que no se pueda saber la cadena original. Para complicar más la contraseña, se le añade un **salt**, es decir, una **serie de caracteres que se añaden de forma aleatoria**, de manera que **dos contraseñas iguales no den el mismo resultado una vez encriptadas**.

En sistemas Linux, la **contraseña encriptada** se solía guardar en el campo **x** del fichero **/etc/passwd**, utilizando un **algoritmo de encriptación** llamado **DES** (**Data Encryption Standard**, algoritmo estándar de encriptación), que encripta las contraseñas utilizando la **función crypt** de Linux. Por motivos de seguridad, se empezó a utilizar el fichero **/etc/shadow**, y algoritmos de encriptación más complejos. A la forma de utilizar este fichero para almacenar las contraseñas se le llama **shadow password**.

Los algoritmos de encriptación más utilizados en Linux son DES (ahora en desuso por su poca seguridad), MD5, SHA-256 y SHA-512. Para saber cuál es, en el archivo **shadow**, en el campo de contraseña, miramos los caracteres de la contraseña y así sabremos el algoritmo utilizado:

- ✔ **DES**: son 13 caracteres, de los cuales los dos primeros son el valor del salt.

- ✔ **MD5**: si empieza por **1**, el valor del salt está entre el segundo y tercer carácter "$" de la cadena, al igual que en los dos siguientes.

- ✔ **SHA-256**: si empieza por **5**.

- ✔ **SHA-512**: si empieza por **6**.

Cada línea del fichero tiene la siguiente estructura:

nombre:contraseña:últ_cambio:mín:máx:aviso:inactividad:exp:reservado

Donde cada campo tiene el siguiente significado:

- ✔ **login**: nombre del usuario.
- ✔ **contraseña**: contraseña del usuario cifrada.
- ✔ **últ_cambio**: fecha última en que se cambió la contraseña.
- ✔ **mín**: mínimo número de días antes de poder volver a cambiar la contraseña.
- ✔ **máx**: máximo número de días que la contraseña puede estar activa. Se utiliza para obligar al usuario a que cambie la contraseña.
- ✔ **aviso**: el número de días de antelación con el que se avisará al usuario de la expiración de su contraseña.
- ✔ **inactividad**: cantidad de días de inactividad, es decir, sin entrar al sistema que puede estar la cuenta antes de que expire.
- ✔ **exp**: fecha en que expira la cuenta. Se cuenta en días a partir del 01-01-1970. Si el campo está en blanco, la cuenta no expirará nunca. Si ha expirado, el superusuario o administrador del sistema deberá rehabilitar la cuenta.
- ✔ **reservado**: campo reservado para usos en el futuro.

Desde el terminal y utilizando el comando **mkpasswd**, generamos contraseñas encriptadas para una misma cadena de caracteres. Utilizaremos el comando varias veces para el mismo algoritmo para comprobar que las contraseñas generadas son diferentes.

Nos muestra los algoritmos que podemos usar.
```
paco@ubuntupaco:~$ mkpasswd -m help
```

Nos pide la contraseña y nos devuelve su valor encriptado.
```
paco@ubuntupaco:~$ mkpasswd -m des
```

Como el salt es aleatorio, si repetimos el comando con la misma cadena nos devuelve valores distintos.
```
paco@ubuntupaco:~$ mkpasswd -m md5
paco@ubuntupaco:~$ mkpasswd -m sha-256
paco@ubuntupaco:~$ mkpasswd -m sha-512
```

Elegimos un salt fijo y al repetir el comando, con la misma contraseña, nos devuelve el mismo valor.
```
paco@ubuntupaco:~$ mkpasswd -m md5 -S 12345678
paco@ubuntupaco:~$ mkpasswd -m sha-256 -S 1234567812345678
paco@ubuntupaco:~$ mkpasswd -m sha-512 -S 1234567812345678
```

/etc/group

Fichero donde se encuentran los grupos definidos en el sistema. Cada usuario del sistema debe pertenecer obligatoriamente a un grupo principal o primario. El grupo principal de cada usuario es el grupo cuyo GID viene en el fichero /etc/passwd. Además, un usuario puede pertenecer a otros grupos, llamados grupos secundarios. No es obligatorio, aunque puede convenir para ciertos casos. Cada línea del fichero tiene la siguiente estructura:

grupo:x:GID:usuarios

Donde cada campo tiene el siguiente significado:

- ✔ **nombre**: contiene el nombre del grupo.
- ✔ **x**: se utilizaba sobre todo en los sistemas para almacenar la contraseña del grupo. Actualmente no se suele utilizar, salvo para proteger ciertos grupos de que un usuario que no pertenezca a él pueda convertirse en miembro del grupo con el **comando newgrp**, para lo que se añade * o ! en el fichero /etc/gshadow, o bien, se puede añadir una contraseña con el **comando gpasswd**, en este caso para permitir que un usuario que no pertenezca al grupo se haga **miembro temporal con newgrp**.
- ✔ **GID**: tiene el **mismo significado** que en el **fichero passwd**. Es el número que identifica al grupo en el sistema.
- ✔ **usuarios**: lista separada por comas de usuarios que tienen a ese grupo como grupo secundario. Para saber si algún usuario tiene el grupo como primario deberíamos mirarlo en el fichero passwd.

/etc/gshadow

Fichero donde se guardan las contraseñas de los grupos del sistema. Aunque las contraseñas no se utilicen para los grupos, es necesario el fichero para proteger el grupo. Al igual que shadow, s**olo root tiene permiso de lectura** sobre el fichero. Cada línea del fichero tiene la siguiente estructura:

nombre:contraseña:administradores:miembros

- ✔ **nombre**: nombre del grupo.
- ✔ **contraseña**: puede ser una contraseña encriptada o bien los caracteres * o !, dependiendo de si queremos usar las contraseñas de grupo o no.

Desde el terminal, miramos en el **subdirectorio /etc** quién es el usuario propietario y el grupo de los ficheros passwd, shadow, group y gshadow.

Vamos al subdirectorio /etc.
```
paco@ubuntupaco:~$ cd /etc
```

Mostramos los archivos en formato largo.
```
paco@ubuntupaco:~$ ls -l passwd *shadow group
```

Volvemos a nuestro subdirectorio personal.
```
paco@ubuntupaco:~$ cd
```

Otros ficheros y directorios de configuración

/etc/default/useradd
Contiene los valores por defecto a la hora de añadir un usuario al sistema con el comando **useradd**.

/etc/adduser.conf
Contiene los valores por defecto cuando se añaden usuarios con el comando **adduser**.

/etc/deluser.conf
Contiene los valores por defecto cuando se eliminan usuarios con el comando **deluser**.

/etc/login.defs
Si se utiliza el sistema de contraseñas encriptadas, este archivo define algunos valores por defecto sobre las encriptación de contraseñas y otros parámetros al generar un usuario nuevo.

Contendrá el algoritmo utilizado para la encriptación de la contraseña, **ENCRYPT METHOD SHA512**, si vamos a permitir o no usar la encriptación con el algoritmo MD5, **MD5_CRYT_ENAB no**, establecer algunas variables del sistema ($PATH), o bien indicar parámetros por defecto para la contraseña, como el número máximo de días que puede ser usada, número mínimo de días entre un cambio y otro, cuándo te debe avisar antes de que expire, etc.

Todos estos parámetros, el usuario root los podría modificar con las opciones de los comandos para la gestión de usuarios como passwd, usermod, o modificando directamente el fichero /etc/shadow.

Buscamos dentro del fichero **/etc/login.defs** la cadena de caracteres “encrypt method”. Hacemos que la encuentre tanto si se escribe en mayúsculas como en minúsculas. Después buscamos la cadena “md5*”.

```
paco@ubuntupaco:~$ grep -i "encrypt method" /etc/login.defs
paco@ubuntupaco:~$ grep -i "md5" /etc/login.defs
```

La **opción -i** hace que **no diferencie entre mayúsculas y minúsculas**.

/etc/shells
El fichero de shells contiene una lista de shells válidos. Si el shell de usuario no está en este fichero, o bien el usuario tiene el **shell /bin/false** y no puede acceder al sistema mediante login.

/etc/skel
Directorio que contiene el **contenido del directorio de los nuevos usuarios** que se vayan añadiendo al sistema.

3. Gestión de usuarios y grupos

En primer lugar, abriremos un terminal en modo texto, o bien podemos usar las terminales virtuales a las que podemos acceder pulsando las teclas CTRL+ALT+F1 hasta CTRL+ALT+F6.

Editor de ficheros

Para ver el contenido de los ficheros de configuración, podemos usar los comandos como **cat**, **more**, **less**, **head** o **tail** si solo queremos ver una parte. Pero, si además, queremos cambiar algún valor de ellos si tenemos permiso para ello, debemos usar un editor de textos. En el modo gráfico, cada vez que abrimos un fichero de texto, lo podemos hacer con el editor **gedit**, pero en modo texto, contamos además con **nano**, **vi**, y **cualquier otro que queramos instalar**, ya que hay bastantes.

- ✔ **gedit**. Es el **editor de textos por defecto en modo gráfico**. Es muy sencillo de usar. Tiene una barra de menús y otra de herramientas. En la de menús, podemos encontrar Archivo, con las opciones, entre otras, de Nuevo, Abrir, Guardar, Guardar como, Vista preliminar de impresión, Imprimir, Cerrar y Salir. Además de las opciones típicas de los editores de texto, que se encuentra en los otros menús, como Buscar, Reemplazar, Cortar, Copiar, Pegar, y Comprobar ortografía, entre otras. Además permite tener más de un documento abierto.

- ✔ **nano**. Es el **editor por defecto en modo texto**. Para ejecutarlo, en la terminal de texto escribimos **nano y el nombre del fichero**. Tiene algunas funciones que se pueden utilizar mediante las combinaciones de teclas que se pueden ver en la imagen. El **caracter** ^ significa que hay que **pulsar la tecla de Con trol (CTRL) junto con la tecla correspondiente a la función** que deseemos.

- ✔ **vi**. Este editor de textos está **integrado en los sistemas operativos** desde Unix. Es más complicado de usar que el anterior, ya que se necesita saber los comandos para utilizarlo. Todavía viene instalado con la versión **vim, que es vi mejorado**.

  Para **empezar a escribir**, deberemos **pulsar la tecla i o la tecla a** para entrar en modo inserción de texto. Después de escribir, para salir, deberemos pulsar **ESC:wq** para guardar el archivo y salir. Si queremos salir sin guardar, deberemos escribir **ESC:q** o bien **ESC:q!** si hemos hecho algún cambio y queremos descartarlo.

Creamos los ficheros **fich1.txt** con **gedit**, **fich2.txt** con **nano** y **fich3.txt** con **vi**. Los guardamos en nuestro directorio personal. Como contenido de los ficheros escribimos fichero 1, fichero2 y fichero3, respectivamente. Después, miramos el contenido de los tres ficheros, usando caracteres comodines.

Creamos el **fichero fich1.txt**.
```
paco@ubuntupaco:~$ gedit fich1.txt
fichero1
Guardar y Salir
```

Creamos el **fichero fich2.txt**.
```
paco@ubuntupaco:~$ nano fich2.txt
fichero2
CTRL+W y CTRL+X
```

Creamos el **fichero fich3.txt**.
```
paco@ubuntupaco:~$ vi fich3.txt
i
fichero3
ESC:wq
```

De la primera forma, sólo mostraría los ficheros que indicamos al principio. De la segunda forma los mostraría, pero también mostraría fich4.txt, ficha.txt, fich5.txt... si existieran.
```
paco@ubuntupaco:~$ cat fich[123].txt
paco@ubuntupaco:~$ cat fich?.txt
```

Ejecutar comandos con privilegios de administrador

su
Cambia de usuario o nos permite ser el superusuario o root.

Sintaxis:
su [-][usuario]

Opciones:
Esta opción se utiliza **para entrar como el usuario que indiquemos, o root si no indicamos ninguno**, ejecutando sus scripts de inicio del usuario.

Argumentos:
usuario
Entra como el usuario que le indiquemos como argumento. Si no se indica ninguno asume que el usuario es root. En todo caso nos pedirá la contraseña del usuario, a menos que sea root el que ejecute el comando.

sudo
Permite **ejecutar comandos como si fuéramos el superusuario**. Previamente, el usuario deberá estar configurado en el **fichero /etc/sudoers** como usuario que puede tener privilegios de root, o bien pertenecer al grupo que se especifique en ese fichero.

Sintaxis:
sudo comando

Argumentos:
comando
Es el comando que ejecutaremos con los privilegios de superusuario. Nos pedirá nuestra contraseña antes de ejecutarlo como medida de seguridad.

Durante un tiempo, después de introducir la contraseña podremos ejecutar otros comandos con privilegios de root sin que nos la vuelva a pedir. También podemos configurar el fichero /etc/sudoers para poder ejecutar comandos con sudo y que no nos pida la contraseña.

Vamos a cambiar la contraseña al usuario root. Entraremos como root y, como root, examinaremos el contenido de los ficheros /etc/shadow y /etc/gshadow de forma paginada por pantalla. Examinaremos también el contenido del fichero /etc/sudoers, y comprobaremos quién tiene o puede tener todos los privilegios en el sistema.

Cambiamos la contraseña a root y entramos como root.
```
paco@ubuntupaco:~$ sudo passwd root
paco@ubuntupaco:~$ su
```

Vemos que es el usuario root y los usuarios que pertenecen al grupo admin.
```
paco@ubuntupaco:~# less /etc/shadow
paco@ubuntupaco:~# more /etc/gshadow
paco@ubuntupaco:~# cat /etc/sudoers
```

Comandos para la gestión de usuarios y grupos

Vamos a ver los diferentes comandos que podemos usar para la **gestión de usuarios y grupos**. Hay que tener en cuenta que solo el administrador o un usuario con privilegios de administrador pueden añadir, eliminar o modificar la configuración de los usuarios y grupos, por lo que desde el terminal tendremos que entrar como root, o bien, como usuario, utilizar el comando sudo.

adduser
Añade usuarios al sistema, también añade un usuario existente a un grupo que también exista previamente en el sistema. Existe también el comando **useradd**, aunque este añade también usuarios después habría que añadirle una contraseña, ya que añade un (!) en el fichero /etc/shadow y crearle el directorio en home, o bien especificárselo en las opciones.

Sintaxis:
adduser [opciones] usuario
adduser [opciones] usuario grupo

Opciones:
--ingroup nomgrupo → crea al usuario con nomgrupo como grupo principal.
--gid num_gid → igual que la opción anterior pero usando el GID del grupo en vez del nombre.
--home directorio → hace que directorio sea el directorio personal del usuario, en vez del directorio por defecto, que sería /hom/nombre_usuario.

Argumento/s:
usuario
Añade el usuario al sistema con el nombre que se le indique.
grupo
Añade el usuario, que debe existir previamente, al grupo, que también debe existir, como grupo secundario.

deluser
Elimina un usuario. Existe también el comando userdel. Hay que tener en cuenta que no se puede borrar un usuario que esté conectado al sistema en ese momento.

Sintaxis:
deluser [opciones] usuario

Opciones:
--remove-home → borra el subdirectorio personal del usuario (con userdel esta opción sería -r).
--remove-all-files → elimina todos los ficheros que pertenezcan al usuario, además de su directorio.

Argumento:
usuario
Elimina el usuario que tenga el nombre que se le indique.

addgroup
Añade un grupo al sistema. Se obtiene el mismo resultado que si usáramos: **adduser --group grupo**.

Sintaxis:
addgroup grupo

Argumento:
grupo
Añade el grupo con el nombre que se le indique como argumento.

delgroup
Elimina un grupo. Se obtiene el mismo resultado que si usáramos: **deluser --group grupo**. Hay que tener en cuenta que no se puede borrar un grupo que esté asignado a un usuario como su grupo primario.

Sintaxis:
delgroup grupo

Argumento:
grupo
Elimina el grupo con el nombre que se le indique como argumento.

usermod
Modifica un usuario ya creado.

Sintaxis:
usermod [opciones] usuario

Opciones:
-d directorio [-m] → cambia el directorio personal.

-m → mueve el contenido del antiguo.
-g grupo → cambia el grupo primario del usuario.
-u uid → cambia el uid del usuario.
-l nombre → cambia el nombre del usuario.
-s shell → cambia el shell.

Argumento:
usuario
Elimina el grupo con el nombre que se le indique como argumento.

groupmod
Modifica un grupo ya creado.

Sintaxis:
groupmod [opciones] grupo

Opciones:
-n nombre → cambia el nombre del grupo.
-g gid → cambia el gid del grupo.

Argumento:
grupo
Elimina el grupo con el nombre que se le indique como argumento.

gpasswd
Añade o elimina usuarios de un grupo. También se utiliza para añadir contraseñas a un grupo.

Sintaxis:
gpasswd [opciones] usuario grupo
gpasswd grupo

Opciones:
-a → añade el usuario al grupo.
-d → elimina el usuario del grupo.
-M → añade varios usuarios al grupo.

Argumentos:
usuario
Es el usuario, o la lista de usuarios (con la opción -M) que se añadirán al grupo.
grupo
Es el grupo al que se le va a añadir los usuarios. Si aparece sin opciones y como argumento único, será el grupo al que se le va a añadir una contraseña.

id
Muestra información sobre el usuario, como el UID, el nombre, los grupos a los que pertenece y los GID de los mismos.

Sintaxis:
id [opciones] [usuario]

Opciones:
-u | --user → escribe solo el UID del usuario.

-n | --name → escribe el nombre del usuario, en vez del UID.

Argumento:
usuario
Es el usuario del que queremos ver la información. Si no se especifica ninguno, se entiende que es el usuario que ejecuta el comando.

groups
Muestra el **nombre de todos los grupos a los que pertenece el usuario**.

Sintaxis:
groups [usuario]

Argumento:
usuario
Es el usuario del que queremos ver la información. Si no se especifica ninguno, se entiende que es el usuario que ejecuta el comando.

chfn
Cambia la información del usuario, que después aparecerá en el campo GECOS del fichero /etc/passwd.

Sintaxis:
chfn [opciones] usuario

Opciones:
-f nombre → cambia el nombre.
-t oficina → cambia el número de oficina.
-w teléfono → cambia el teléfono del trabajo.
-h teléfono → cambia el teléfono del domicilio.
-o otro → cambia otra información adicional.

Argumento:
usuario
Es el usuario del que queremos cambiar la información. Si no se especifica ninguno, se entiende que es el usuario que ejecuta el comando.

finger
Muestra información sobre el usuario que se le indique. Si no se especifica ninguno, muestra información sobre los usuarios conectados al sistema.

Sintaxis:
finger [usuario]

newgrp
Se utiliza para **cambiar el grupo primario de un usuario**, hasta que se vuelva a cambiar o finalicemos la sesión. Si queremos cambiar a un grupo al que no pertenezcamos, nos pide la contraseña del grupo.

Sintaxis:
newgrp grupo

chsh
Comando para **cambiar el shell de un usuario**. Si no se especifica ningún usuario se entiende que es el usuario que ejecuta el comando.

Sintaxis:
chsh [opciones] [usuario]

Opciones:
-s shell → cambia el usuario al shell que se indique.

Desde el terminal, creamos un grupo nuevo llamado gruponuevo. Después, añadimos un usuario llamado usunuevo, cuyo grupo primario o principal será gruponuevo. A continuación, comprobamos lo realizado y creamos otro grupo llamado grupo2, donde hacemos que usu2 pertenezca a grupo2, pero como grupo secundario. Finalmente, volveremos a comprobarlo todo.

```
paco@ubuntupaco:~$ sudo addgroup gruponuevo
paco@ubuntupaco:~$ sudo adduser --ingroup gruponuevo usunuevo
paco@ubuntupaco:~$ id usunuevo
```

Entramos como usunuevo para ver si está creado.
```
paco@ubuntupaco:~$ su usunuevo
paco@ubuntupaco:~$ whoami
paco@ubuntupaco:~$ exit

paco@ubuntupaco:~$ sudo adduser --group grupo2
paco@ubuntupaco:~$ sudo adduser usunuevo grupo
```

Miramos el archivo de todos los grupos creados.
```
paco@ubuntupaco:~$ cat /etc/group
```

Miramos los grupos a los que pertenece usunuevo.
```
paco@ubuntupaco:~$ groups usunuevo

paco@ubuntupaco:~$ su usunuevo
paco@ubuntupaco:~$ exit

paco@ubuntupaco:~$ sudo deluser --remove-home usunuevo
paco@ubuntupaco:~$ sudo delgroup grupo2
paco@ubuntupaco:~$ sudo deluser --group gruponuevo
```

Comandos para cambiar los ficheros y directorios de usuario y grupo

En Linux, todos los archivos pertenecen a un usuario, que es el propietario del mismo, y a un grupo al que pertenezca el usuario propietario, pero se puede cambiar el propietario y el grupo del archivo. Para ello, podemos usar los siguientes comandos:

chown
Cambia el propietario del archivo. También lo podemos usar para cambiar el grupo.

Sintaxis:
chown [opciones] [propietario[:grupo]] archivo/s

Opciones:
-r | --recursive → cambia el propietario y el grupo, si se le especifica, a un árbol de directorios.

Argumento:
propietario
Es el nuevo propietario al que se le va a asignar el archivo.
grupo
Si queremos cambiar también el grupo.
archivo/s
Es el archivo o los archivos a los que se les van a cambiar el propietario y/o el grupo.

chgrp
Cambia el grupo del archivo.

Sintaxis:
chgrp [opciones] grupo archivo/s

Opciones:
-r | --recursive → cambia el grupo a un árbol de directorios.

Argumento:
grupo
Es el grupo que se le va a asignar al archivo.
archivo/s
Es el archivo o los archivos a los que se les van a cambiar el grupo.

4. Permisos

Al existir diferentes usuarios y grupos y como todos los archivos deben pertenecer a un usuario y a un grupo, se hace necesario protegerlos utilizando tres grupos de permisos. Los permisos del propietario del archivo, los permisos de los usuarios que pertenecen al grupo al que pertenece el archivo, y el resto de los usuarios.

Los permisos más frecuentes que se pueden añadir al archivo son de **lectura** ("r"), **escritura** ("w") y **ejecución** ("x"). Si se trata de un directorio, un permiso de ejecución significa que podemos entrar en él. Si es un fichero, el permiso de ejecución tendrá significado si el fichero es ejecutable o binario, o bien contiene otros comandos, es decir, es un shell script.

chmod
Cambia los permisos de un archivo.

Sintaxis:
chmod [permisos[,permisos]...] archivo/s

Permisos:
Los permisos se pueden añadir o quitar indicando lo siguiente:
chmod {a, u, g, 0} {+ | - } {r, w, x, X, s, t} archivos

La primera letra indica a quién se le va a cambiar el permiso: al **propietario** (**u**), a los del **grupo** (**g**), a los **demás** (**o**) o a **todos** (**a**), que es la opción por defecto.

Después podemos **añadir** (+) o **quitar** (-) el permiso.

Por último, indicamos los permisos que le queremos dar o quitar: **lectura** (**r**), **escritura** (**w**), **ejecución** (**x**), **ejecución en caso de que sea un directorio** (**X**).

Existen **dos permisos especiales**:
- s (**setuid bit** o **setgid bit**), según se aplique a un usuario o grupo, indica que el proceso pertenece al dueño del programa o a su grupo, no al que lanzó el proceso.
- t (**sticky bit**), que aplicado a directorios hace que si un usario tiene permiso de escritura, puede modificar los archivos, pero no los puede mover ni borrar.

Los permisos también se pueden indicar de forma numérica, como un **número octal de hasta 4 dígitos**. El primer dígito de la derecha indica los permisos que se les va dar a los demás; el segundo, permisos del grupo; el tercero, permisos al usuario y el cuarto, **si hay cuarto número, permisos especiales**.

Los números indican:
0: ningún permiso
1: permiso de ejecución
2: permiso de escritura
4: permiso de lectura

Para asignar **combinaciones de permisos podemos sumar los números**.

En el caso del número de los permisos especiales, indican:
1: sticky bit
2: setgid
4: setuid

Argumento:
archivo/s
Es el archivo o los archivos a los que se les van a cambiar los permisos.

umask
Cambia o te muestra los permisos que se asignarán por defecto.

Sintaxis:
umask [máscara_de_permisos]

Argumento:
La máscara de permisos tendrá **distintas consecuencias** dependiendo de si después de establecerla **se crea un fichero o un directorio**. Existen permisos base para los ficheros, que es 666, y para los directorios, que es 777.

Partiendo de esta base de permisos:
- Para los **directorios** podemos **restarle el umask a los permisos base**.
- Para los **archivos** hacemos AND de la negación de umask, con lo que **el último bit no se activa nunca**.

Si umask tiene el valor de 033, ¿con qué permisos se crearía un fichero y un directorio?

Comando umask 033	→ `000 011 011`
Negación de umask (744)	→ `111 100 100`
Base de permisos para ficheros (666)	→ `110 110 110`
Negación AND base de ficheros	→ `110 100 100`

Permisos que obtenemos para un **fichero** → **644 (rw- r-- r--).**
Permisos que obtenemos para **directorios** → **777 – 033 = 744 (rwx r-- r--).**

Debemos recordar la **tabla de verdad de AND** para poder realizar la operación bit a bit.

Examinamos el valor de umask y comprobamos los permisos de las entradas de tu directorio personal. Después, cambiamos el valor de umask a 000. Creamos un directorio llamado direcnuevo y un fichero llamado ficheronuevo. Comprobamos los permisos que tienen. Ahora, los cambiamos de manera que en ambos nosotros (el usuario propietario) tenga todos los permisos activos, los de tu grupo puedan leer y ejecutar, y los otros no tengan ningún permiso activo.

```
paco@ubuntupaco:~$ ls -l
paco@ubuntupaco:~$ umak 000
paco@ubuntupaco:~$ mkdir direcnuevo
paco@ubuntupaco:~$ nano ficheronuevo
paco@ubuntupaco:~$ ls -ld *nuevo
paco@ubuntupaco:~$ chmod 750 direcnuevo
paco@ubuntupaco:~$ chmod u+x, g-w, o-rw ficheronuevo
```