

## Unidad de trabajo 9: Gestión de usuarios y grupos.

Como ya sabemos Linux soporta el concepto de múltiples usuarios y grupos. Un usuario es alguien que tiene un identificador único en el sistema. Un grupo es un conjunto de usuarios y también cada grupo tiene una identificación única.

### **Tipos de usuarios:**

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

- ♦ **Usuario root.** También se 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, de la actualización y del mantenimiento. Se identifica porque 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 tiene todos los privilegios, tienen algunos dependiendo del usuario de que se trate. No son usuarios que entran al sistema e inician la sesión. El UID que tienen es mayor que 1 y menor que 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.

Todo usuario posee información periférica adjunta.

<b>INFORMACIÓN RELATIVA AL USUARIO</b>	
<b>CAMPO</b>	<b>DESCRIPCIÓN</b>
Login /Nombre de Usuario	Nombre único del usuario en el sistema
Password	Contraseña en virtud de la cual el usuario puede acceder al sistema
UID	Abreviatura de identificación del usuario. Este número se empareja con el nombre login. Toda combinación Login/UID debe ser única
GID	Abreviatura de identificación de grupo. Todos los usuarios pertenecen al menos a un grupo.
Comentario	Es el nombre completo del usuario e Información de contacto.
Ruta del directorio de inicio	Es el directorio donde están almacenados los archivos de usuario
Shell	Es el intérprete de comandos que se inicia cuando el usuario comienza la sesión

## 1. AÑADIR UN USUARIO.

Nombre completo: Juan José que trabaja en el Dpto. Comercial y el Teléfono del trabajo es 928181818.

Shell de conexión: /bin/bash; directorio personal : /home/Juanjo y la contraseña: juanito

Ver la entrada que se crea en el archivo /etc/passwd **#cat /etc/passwd**

juanjo:x:1003:100:Juanito Pérez,Dpto.Comercial,928181820,,:/home/juanjo:/bin/bash ,  
donde:

**x:** significa que está siendo usada el sistema de claves shadow, que mantendrá las claves encriptadas en el fichero /etc/shadow

**1003** es el id del usuario (UID)

**100** (GUID) es el grupo al que pertenece el usuario (es la abreviatura de identificación de grupo users) nos lo proporciona el sistema,

**Juanito Pérez,Dpto. Comercial,928181820**(comentario) es el nombre completo e información de contacto.

**/home/juanjo** es (directorio de inicio) el directorio personal y **/bin/bash** (shell) intérprete de comandos que utilizará el usuario.

Ver la entrada que se crea en el archivo /etc/shadow **#cat /etc/shadow**

juanjo:\$1\$EqEr0\$MfhVM2wxmmKuD5VkYKzZz1:14017:0:99999:7:::

(juanjo fue creado en 18 de mayo de 2008) (Aquí es donde se registra la clave encriptada del usuario)

nuevo:\$6\$.0MnuXk/\$NOxfzfz9B.wzvvnvQReyFQf8e0LLkyY7hdLOWRau5IvzWdepCPpUfMpi  
MvMANZAE2wH7LI2bG5Xj3KaP7UkoRq/:15845:0:99999:7:::

(Nuevo fue creado el día 20 de mayo de 2013)

juanjo2:\$1\$S6/tALCn\$o.5h104WY11JQgBVJguLa1:2:3:15:7:::

(juanjo2 creado el 3 de enero de 1970)

Si en lugar de la clave encriptada aparece un \* o una exclamación quiere decir que el usuario no tiene clave asignada.

**2:** Fecha del último cambio de clave (Contada en días a partir de 1 de enero de 1970)

**3:** La contraseña no puede cambiarse antes de 3 días, después del último cambio de contraseña (Si el último cambio fue el 03/01/1970, el usuario la podrá cambiar, el 06/01/1970)

**15:** La contraseña expira 15 días después del último cambio de contraseña (el 18 de enero de 1970, no puede entrar)

**7:** Cantidad de días preaviso de expirar clave

**Primer campo en blanco:** cantidad de días entre expiración de la clave y bloqueo de cuenta

**Segundo campo en blanco:** Fecha desde la cual la cuenta está deshabilitada (contada en días desde el 1 de enero 1970)

slice:\$1\$NLJJ6\$ow5g1I1NgYITqqQQy5D21:14234:0:99999:7: : :		
Contraseña	Caducidad	Días a los que se deshabilita la cuenta contados desde el 1 de enero de 1970.
	Inactivo	Días a los que se deshabilita la cuenta después de que caduque la contraseña.
	Aviso	Días a los que el usuario será avisado de que debe cambiar la contraseña antes de que ésta caduque.
	Máximo	Días durante los que la contraseña es válida. Al terminar el usuario tiene que cambiar la contraseña.
	Mínimo	Días que deben pasar como mínimo para que el usuario pueda cambiar la contraseña.
	Último cambio	Días que han pasado desde la última vez que la contraseña fue cambiada contados desde el 1 de enero de 1970.
Contraseña	Contraseña encriptada. La forman entre 13 y 24 caracteres (a-z, A-Z, 0-9, \, /). Si comienza por el carácter \$, inidca que la contraseña se ha encriptado usando un algoritmo distinto de DES. Si comienza por \$1\$, el algoritmo de cifrado está basado en MD5.	
Nombre de usuario	Nombre que identifica al usuario en el sistema. Debe tener entre 1 y 32 caracteres.	

## 2. AÑADIR UN USUARIO MEDIANTE UN COMANDO

```
#useradd -m -c "Pedro Pérez, Dpto Informática, 928928928,938638638" -s /bin/bash
-u 2010 -p pedrito pedrito
```

useradd o adduser es el comando que permite añadir nuevos usuarios al sistema desde la línea de comandos. Sus opciones más comunes o importantes son las siguientes:

- -c añade un comentario al momento de crear al usuario, campo 5 de /etc/passwd
- -d directorio de trabajo o home del usuario, campo 6 de /etc/passwd
- -e fecha de expiración de la cuenta, formato AAAA-MM-DD, campo 8 de /etc/shadow
- -g número de grupo principal del usuario (GID), campo 4 de /etc/passwd
- -s shell por defecto del usuario cuando ingrese al sistema. Si no se especifica, bash, es el que queda establecido.
- -u UID del usuario, si no se indica esta opción, automáticamente se establece el siguiente número disponible a partir del último usuario creado.
- -m : Crear una carpeta home si es que no existe.
- -p: password encriptada

Ahora bien, realmente no hay prácticamente necesidad de indicar ninguna opción ya que si hacemos lo siguiente:

```
#> useradd juan
```

Se creará el usuario y su grupo, así como las entradas correspondientes en /etc/passwd, /etc/shadow y /etc/group. También se creará el directorio de inicio o de trabajo: /home/juan y los archivos de configuración que van dentro de este directorio y que más adelante se detallan.

Las fechas de expiración de contraseña, etc. Quedan lo más amplias posibles así que no hay problema que la cuenta caduque, así que prácticamente lo único que faltaría sería añadir la contraseña del usuario y algún comentario o identificación de la cuenta. Como añadir el password o contraseña se estudiara en un momento y viendo las opciones con '-c' es posible establecer el comentario, campo 5 de /etc/passwd:

```
#> useradd -c "Juan Perez Hernandez" juan
```

Siempre el nombre del usuario es el último parámetro del comando. Así por ejemplo, si queremos salirnos del default, podemos establecer algo como lo siguiente:

```
#> useradd -d /usr/juan -s /bin/csh -u 800 -c "Juan Perez Hernandez" juan
```

Con lo anterior estamos cambiando su directorio de inicio, su shell por default sera csh y su UID será el 800 en vez de que el sistema tome el siguiente número disponible.

**Ver las entradas en los archivos correspondientes.**

### ELIMINAR UN USUARIO

```
#userdel -r juanjo
```

Elimina el usuario y los archivos del directorio /home/usuario1  
Comprobar entradas en los ficheros correspondientes.

Vuelve a crear el usuario juanjo con las propiedades del punto 1 desde la línea de comandos.

### 3. MODIFICAR UN USUARIO

Podemos cambiar algunas de las características del usuario mediante el entorno gráfico. También lo podemos hacer con el comando **usermod** y los siguientes parámetros:

- c nombre\_completo (comentario)
- g grupoinicial
- G grupos a pertenecer
- l nuevo\_login
- e cambia la expiración de la cuenta. Se desactiva la cuenta
- p cambia la contraseña.

Ejemplos:

Modificar el nombre por "Pedro Melián" **#usermod -c "Pedro melián" pedrito**

Modificar el login de pedrito por pedrillo **#usermod -l pedrillo pedrito**

**Modificar el login nuevamente por pedrin y comprobar entrada.**

### 4. BLOQUEAR UN USUARIO. DESACTIVAR TEMPORALMENTE

Puede llegar el momento de que un usuario sea problemático dentro del sistema y que se tenga que bloquear su acceso. No se trata de borrar su cuenta del sistema sino de desactivarlo por un tiempo. La acción que resulta más fácil es como operador root cambiarle la contraseña, sin previo aviso.

## GRUPOS

Todo grupo posee una información periférica adjunta

INFORMACIÓN RELATIVA AL GRUPO	
CAMPO	DESCRIPCIÓN
Name	Es el nombre del grupo, que no debe ser un nombre compuesto
Password	Si un grupo tiene contraseña deberá estar configurado. Casi nunca tendremos que preocuparnos por la configuración
GID	Abreviatura de identificación del grupo. Esto asocia un número al nombre del grupo. Todas las combinaciones nombre/grupo deben ser únicas
Lista de usuario	Usuarios incluidos dentro del grupo

Todos los usuarios están asociados al menos con un grupo. Un grupo es una colección de usuarios unidos con un propósito.

Cuando creamos un usuario podemos indicarle al sistema en qué grupo va a encontrarse el usuario, de hecho al crear un usuario `Useradd -m -c "Pedro Pérez" -s /bin/bash -g 100` pedrito (100 grupo users) la información que nos aparece en el archivo `passwd` es:

```
pedrito:x:1001:100:Pedro Pérez:/home/pedrito:/bin/bash
```

El grupo inicial se refleja en el fichero `passwd`.

## 5. AGREGAR UN GRUPO

A) En entorno gráfico Crear el grupo "grupito" al cual pertenecerán los usuarios juanjo y pedrin. Comprobar entrada en el fichero, donde aparecerá en `/etc/group` aparece una entrada correspondiente de grupito.

```
grupito:x:2013:juanjo,pedrin
```

Eliminar el grupo anterior **#groupdel grupito** y comprobar entrada en fichero

B) Mediante la edición del archivo: `/etc/group`. Añadir la misma entrada anterior basándonos en la siguiente sintaxis:

```
nombre_de_grupo:x:id_grupo:user1,user2
```

Comprobar la entrada en entorno gráfico

C) Mediante línea de comandos con la orden **groupadd # groupadd grupito2** y comprobar entrada.

Ejemplo: crear un grupo que se llame `www` cuyo número de identificación sea el 1007

**#groupadd -g 1007 www** (con este comando si me añade una entrada en el fichero `/etc/gshadow` y `/etc/group` )

## 6. El comando newgrp

Los usuarios pueden pertenecer a varios grupos, linux sólo permite que estén activos en uno de ellos en un momento dado. El grupo inicial, llamado normalmente grupo primario, es el grupo identificado en el archivo `/etc/passwd`. Si el usuario quiere trabajar en otro grupo, deberá utilizar la orden `newgrp`:

Sintaxis: **newgrp - (espacio en blanco) nombre\_del\_grupo**

Si trabajo dentro del grupo 100, los archivos pertenecen al grupo 100. Si trabajo dentro del grupo 1000 los archivos pertenecerán al grupo 1000. Siempre que estemos en grupo secundario y queramos volver al grupo inicial basta con poner **newgrp**.

## 7. El comando id

Para saber sobre que grupo estamos trabajando: **id -g**, (id sin parámetros, es para indicar entre otras cosas los grupos a los que pertenece un usuario)

## 8. MODIFICAR UN GRUPO EXISTENTE

A) Editando el archivo `/etc/group`, puedo cambiar el nombre, el número de identificación y el conjunto de usuarios asociados a él.

B) Con el comando **groupmod**. Nosotros tenemos el grupo que se llama `www` y con el número de identificación 1007. Vamos a cambiarle el nombre a `wwwwww` y el número de identificación 1008.

```
#groupmod -g 1008 -n wwwwww www
```

## 9. ELIMINAR UN GRUPO EXISTENTE

A) Editando el archivo `/etc/group`, elimino la entrada correspondiente al grupo

B) Utilizando el comando **groupdel**. Vamos a eliminar el grupo `pepita`. Para ello ejecutamos `#groupdel wwwwww`

## 10. UTILIZAR EL COMANDO chgrp

```
drwxr-xr-x 2   pedrin      users      4096 May  24 16:11   caja
drwxr-xr-x 2   pedrin      grupito    4096 May  24 16:11   cajita
```

En este caso `pedrin` ha creado el directorio `caja` como usuario del grupo `users`. Luego realiza un cambio con `newgrp` a `grupito` para pasar como usuario al otro grupo y ha creado `cajita`. Imaginemos que `pedrin` quiere poner `caja` dentro del grupo `grupito`.

```
$chgrp grupito caja
```

Con `-R` cambia todo lo incluido en la carpeta.

## 11. UTILIZAR EL COMANDO chown

Este comando se utiliza para cambiar el propietario de un archivo o directorio, aunque también puede cambiar el grupo. Para ello debemos estar como administrador del sistema.

Si solo se da un nombre de usuario o UID entonces ese usuario se convierte en el propietario y el grupo al que pertenece el fichero no cambia. Si al nombre de usuario le siguen: y un nombre de grupo o GID, entonces también cambiará el grupo.

Utilizando este comando el archivo o directorio se queda donde está, solo es un cambio de dueño.

```
#chown mike /home/pedrin/cajita (chown 1005 /home/pedrin/cajita)
```

**#chown 1005:1004 /home/pedrin/cajita** (cambia el usuario y el grupo propietario)  
chown con -R es para cambiar de propietario a un directorio y a todo su contenido.

## 12. PERMISOS DE ARCHIVOS Y DE DIRECTORIOS Y COMANDO chmod

La orden chmod nos va a permitir modificar los permisos de un objeto. Para modificarlos debemos ser los propietarios (no hace falta comentar que root si puede cambiarlos)

En linux, todo archivo y directorio tiene tres niveles de permisos de acceso: Los que se aplican al propietario del archivo, los que se aplican al grupo que tiene el archivo, y los que se aplican a todos los usuarios del sistema.

Cada sección posee su propio conjunto de permisos, que permiten leer, escribir o ejecutar o lo impiden.

Existen dos métodos para especificar los modos de permiso de un objeto. Se puede utilizar el sistema de codificación numérica o el sistema de codificación por letras. Para la codificación por letras, las tres secciones se denominan:

u	para usuario
g	para el grupo
o	para el resto
a	para los tres grupos

Hay tres tipos básicos de permisos:

r	read - lectura
w	write - escritura
x	execute - ejecutar

La combinación de estos tres permisos (r,w y x) con las tres secciones crea los permisos de los objetos.

El comando ls -l muestra un listado largo, y proporciona bastante información sobre el objeto.

```
drwxr-xr-x 2  pedrin    users      4096 May  24 16:11  caja
    (caja es un directorio de pedrin, y el grupo es users)
drwxr-xr-x 2  pedrin    grupito    4096 May  24 16:11  cajita
    (cajita es un directorio de pedrin y el grupo es grupito)
-rw-r--r-- 1  root      root        12   May  17 19:52  esct
```

El primer carácter justo antes de los permisos, identifica el tipo de objeto, de los que nos interesa saber

d	directorio
-	archivo normal
l	enlace simbólico

Los tres conjuntos de permisos en caso del fichero son: rw- (para el propietario) r-- (para el grupo que posee el archivo o directorio) y r -- (para el resto)

Para la codificación por números cada atributo tiene un valor, un peso correspondiente: r vale 4, w vale 2 y x vale 1.

Cuando se combinan los atributos, se agregan sus valores

r	w	x	valor	Significado
4	2	1		
-	-	-	0	Sin permisos
-	-	x	1	Permiso de solo ejecución
-	w	-	2	Sin Sentido (*)
-	w	x	3	Sin Sentido(**)
r	-	-	4	Permiso de solo lectura
r	-	x	5	Permiso de lectura y ejecución
r	w	-	6	Permiso de lectura y escritura
r	w	x	7	Todos los permisos

(\*) no tiene sentido que yo tenga permiso de escritura, pero no de lectura, ya que para modificar algo, siempre lo podré visualizar

(\*\*) No tiene sentido por la misma razón.

Cuando se combinan los valores de la anterior tabla, se obtienen tres números que conforman el permiso de archivo. Entre las combinaciones de permisos de archivos destacan las siguientes:

Arc h	r	w	x	r	w	x	r	w	x	Valo r
	4	2	1	4	2	1	4	2	1	
-	r	w	-	-	-	-	-	-	-	600
-	r	w	-	r	-	-	r	-	-	644
-	r	w	-	r	w	-	r	w	-	666
-	r	w	x	-	-	-	-	-	-	700
-	r	w	x	r	-	x	r	-	x	755
-	r	w	x	r	w	x	r	w	x	777
-	r	w	x	-	-	x	-	-	x	711

Entre las combinaciones de permisos de directorios destacan las siguientes:

dir	r	w	x	r	w	x	r	w	x	Valo r
	4	2	1	4	2	1	4	2	1	
d	r	w	x	-	-	-	-	-	-	700
d	r	w	x	r	-	x	r	-	x	755
d	r	w	x	-	-	x	-	-	x	711

Con r-x en grupo puedes acceder pero no modificar, por ejemplo crear un directorio.



Con `rw-` en grupo no puedes acceder al directorio, permiso denegado.

Ahora vamos a realizar algunos ejemplos con `chmod`:

Se puede ejecutar utilizando la codificación numérica:

**`chmod 777 nombre_archivo`**

**`chmod u+w og+rx nombre_archivo`** (permiso de escritura para el propietario, permiso de lectura y ejecución para el grupo y el resto) Para quitar permisos ponemos.

## **OTRAS ÓRDENES RELACIONADAS CON USUARIOS**

**su:** Cambia de usuario o nos permite ser superusuario o root.

**Sintaxis:** `su [usuario]`

Esta opción se utiliza para entrar como el usuario que indicaremos o root si no indicamos ninguno.

**sudo:** Permite ejecutar comandos como si fuéramos el supe usuario. Previamente el usuario deberá estar configurado en el fichero `/etc/sudoers`, como que puede tener privilegios de root. O bien pertenecer al grupo.

Para cambiar la contraseña de root: **`sudo passwd root`** \_\_\_\_\_ cambiamos la contraseña a root

`less /etc/shadow`

`cat /etc/sudoers` \_\_\_\_\_ vemos que el usuario root y los usuarios que pertenecen al grupo admin

**gpaswd:** añade o elimina usuarios de un grupo. También se utiliza para añadir contraseña a un grupo.

**`gpaswd [opciones] usuario grupo`**

**Opciones:**

-a añade el usuario al grupo

-d elimina el usuario del grupo.

-M añade varios usuarios al grupo.

**groups** : muestra el nombre de los grupos a los que pertenece el usuario.

**Sintaxis :** `groups [usuario]`

Si en `[usuario]` no se especifica ninguno se entiende que es usuario que ejecuta el comando

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

Sintaxis: `finger [usuario]`

**logname** muestra el nombre de usuario.

**umask** : cambia o te muestra los permisos que se asignan por defecto.

**Sintaxis:** `umask [máscara de permisos]`

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 de los directorios es 777

Partiendo de esta base de permisos:

Generalmente el valor por defecto es 022 en casi todas las distribuciones linux, puedes comprobarlo con el siguiente comando:

### **Umask**

#### **Directorios**

Permisos base para directorios: 777

Permisos definidos en umask: 022

Operación: Restarle el umask a los permisos base de la siguiente manera:

$$777 - 022 = 755$$

Tenemos que los directorios se crearán con permisos de lectura, escritura y ejecución para el propietario y solo de lectura y ejecución para todos los demás.

#### **Archivos**

Permisos base para archivos: 666

Permisos definidos en umask: 022

Operación: Restarle el umask a los permisos base de la siguiente manera:

$$666 - 022 = 644$$

Tenemos que los archivos se crearán con permisos de lectura y escritura para el propietario y solo de lectura para todos los demás.