

## 1.2. Gestión de usuarios

### 1.2.1. Creación de usuarios

**useradd.** El fichero **/etc/passwd** contiene los usuarios del sistema. Permite añadir un usuario indicando como parámetros la información particular para crear el usuario en la misma línea de comandos. La sintaxis es:

**useradd [opciones] nombre-usuario**

Entre las opciones más destacables tenemos:

- c comentarios** introduce un comentario para tener más información del usuario.
- d home\_directory** directorio de trabajo del usuario. Suele ser `/home/nombre-usuario`.
- s shell** intérprete de comandos (shell) del usuario. Suele ser `/bin/bash`.
- g gid** grupo base del usuario (el grupo debe existir previamente).
- m** crea el directorio home para el nuevo usuario.
- G otros\_grupos** lista de grupos secundarios de los que el usuario es miembro. Los grupos se separan por coma, sin espacio en blanco. Los grupos deben existir. Si el usuario es miembro de un grupo que no esté en esta lista, se borrará del grupo.
- u uid** sirve para especificar un UID al usuario. Debe ser un número positivo y único. Por defecto, el sistema asigna el primer UID libre.

Ejemplo:

Si deseamos crear un usuario llamado “pedro” cuyo grupo principal sea “profesores”, cuya carpeta home sea `/home/pedro` y su intérprete de comandos sea `/bin/bash`, ejecutaremos el siguiente comando:

```
sudo useradd -g profesores -d /home/pedro -m -s /bin/bash pedro
```

De esta manera habremos creado al usuario pedro y su carpeta home. Si no utilizamos la opción -m, no se creará la carpeta home del usuario; en tal caso tendríamos que crearla manualmente.

**Nota:** Se recomienda que el nombre de usuario sea en minúsculas y además de letras también puede contener números y algún signo como guiones normales y guiones bajos. Debemos recordar que unix distingue entre mayúsculas y minúsculas, es decir, “Pepe” es distinto de “pepe”.

Tan solo nos quedará establecer su contraseña y para ello usaremos el siguiente comando.

**passwd.** Permite modificar la clave (password) de un usuario dado. La sintaxis es:

```
passwd [usuario]
```

Ejemplo:

En caso de escribir el comando sin especificar el nombre del usuario, simplemente cambiaremos la clave del usuario actual.

```
passwd
```

Después se recibirá la respuesta:

**Cambiando la contraseña para [nombre usuario actual].**

**(actual) contraseña de UNIX:**

**Introduzca la nueva contraseña de UNIX:**

**Vuelva a escribir la nueva contraseña de UNIX:**

Cada uno de estos datos debe ser introducido correctamente para que la contraseña se cambie satisfactoriamente.

Ejemplo:

En caso de especificar un usuario, es necesario que el comando lo ejecute el usuario root. Siguiendo el ejemplo del comando *useradd*, el ejemplo quedaría:

**passwd pedro**

En este caso no habría que introducir la contraseña actual del usuario “pedro”, sólo habría que introducir dos veces la nueva contraseña.

### 1.2.2. Modificación de usuarios

**usermod.** El fichero **/etc/passwd** contiene los usuarios del sistema. Permite cambiar el nombre del usuario, su carpeta home, su intérprete de comandos, los grupos a los que pertenece y algunos otros parámetros. La sintaxis es:

**usermod [opciones] nombre-usuario**

Entre las opciones más destacables tenemos:

**-c comentarios** modifica el comentario del usuario.

**-d home\_directory** modifica el directorio de trabajo del usuario.

**-s shell** modifica el intérprete de comandos (shell) del usuario.

**-g grupo\_base** modifica el grupo base del usuario (debe existir previamente).

**-G otros\_grupos** lista de grupos secundarios de los que el usuario es miembro. Los grupos se separan por coma, sin espacio en blanco. Los grupos deben existir. Si el usuario es miembro de un grupo que no esté en esta lista, se borrará del grupo.

**-u uid** cambio de identificador de usuario. Este valor debe ser único. No admite valores negativos. Cualquier uid de fichero que esté en el directorio origen del usuario se cambiará automáticamente al nuevo uid, pero no pasa así con los ficheros que estén en otro directorio, que deberán ser cambiados manualmente (usando **chmod**, **chgrp**, **chown**).

**-l nombre\_login** cambio de login.

**-L** bloquea la contraseña, deshabilitando la cuenta de usuario.

**-U** desbloquea la contraseña, habilitando la cuenta de usuario para iniciar sesión.

Ejemplo:

Queremos cambiar el home del usuario "pedro" a */home/carpeta\_pedro* y su grupo base a *grupo\_base\_pedro*:

```
sudo usermod -d /home/carpeta_pedro -g grupo_base_pedro pedro
```

### 1.2.3. Eliminación de usuarios

**userdel.** El fichero **/etc/passwd** contiene los usuarios del sistema. Elimina la cuenta de usuario indicada. La sintaxis es:

**userdel [opciones] nombre-usuario**

Entre las opciones más destacables tenemos:

**-r** elimina el directorio personal del usuario especificado.

Ejemplo:

Queremos eliminar el usuario “pedro” y su carpeta home:

```
sudo userdel -r pedro
```

### 1.2.4. Otros comandos de usuarios

**id.** Muestra el UID, GID del grupo por defecto y los GID de los otros grupos a que pertenece el usuario indicado. Si no se especifica el usuario se utiliza el usuario actual. La sintaxis es:

**id [nombre-usuario]**

Ejemplo:

Queremos conocer el UID y los GIDs del usuario “pedro”:

```
sudo id pedro
```

### 1.3. Gestión de grupos

#### 1.3.1. Creación de grupos

**groupadd.** El fichero **/etc/group** contiene los grupos del sistema. Permite añadir un grupo de usuarios vacío indicando como parámetro el nombre del grupo. La sintaxis es:

**groupadd [opciones] nombre-grupo**

Entre las opciones más destacables tenemos:

- g gid** especifica el GID del grupo que se va a crear.
- o** indica que el grupo puede tener asociado más de un GID.
- r** crea un grupo de sistema.

Ejemplo:

Deseamos crear un grupo llamado 'alumnos' con GID 500:

**sudo groupadd -g 500 alumnos**

#### 1.3.2. Modificación de grupos

**groupmod.** El fichero **/etc/group** contiene los grupos del sistema. Permite modificar el nombre de un grupo o el GID del mismo. La sintaxis es:

**groupmod [opciones] nombre-grupo**

Entre las opciones más destacables tenemos:

- n** cambia el nombre del grupo.
- g** cambia el GID.

Ejemplo:

Cambiamos el nombre del grupo “profesores” a “docentes y su GID a “2000”:

```
sudo groupmod -n docentes -g 2000 profesores
```

### 1.3.3. Eliminación de grupos

**groupdel.** El fichero **/etc/group** contiene los grupos del sistema. Elimina un grupo de usuarios creado previamente. La sintaxis es:

```
groupdel nombre-grupo
```

Ejemplo:

Eliminamos el grupo de usuarios llamado “alumnos”:

```
sudo groupdel alumnos
```

### 1.3.4. Cambiar el grupo de un usuario

**newgrp.** El fichero **/etc/group** contiene los grupos del sistema. Cuando se accede al sistema siempre se hace con el mismo grupo. Pero un usuario puede pertenecer a más de un grupo aunque sólo está activo uno. El usuario puede necesitar cambiarlo por otro de los que pertenezca para crear nuevos ficheros. Para ello usará el comando que nos ocupa. Ha de ser el usuario desde su login quien ejecute esta operación. Su sintaxis es:

```
newgrp nuevo-grupo
```

Ejemplo:

El usuario “maria” tiene como grupo principal el grupo “maria”, pero también pertenece al grupo “admin”. Si en un momento dado quiere crear un archivo en un

directorio al que no tiene permisos para acceder el grupo “maria” pero sí el grupo “admin”, el usuario “maria” puede cambiar su grupo principal por el grupo “admin” y realizar las operaciones concretas.

#### **groups**

sale por pantalla

**maria cdrom dalout admin sambashare**

el grupo con el que está trabajando el usuario “maria” actualmente es el grupo “maria”

**newgroup admin**

#### **groups**

sale por pantalla

**admin cdrom dalout sambashare maria**

el usuario “maria” está trabajando ahora mismo con el grupo “admin”. Ya puede realizar las operaciones que quería llevar a cabo.

### **1.3.5. Añadir usuarios a un grupo**

**adduser.** Permite añadir un usuario del sistema a un grupo que debe existir previamente. La sintaxis es:

**adduser nombre-usuario nombre-grupo**

Ejemplo:

Añadimos el usuario “andres” al grupo “profesores”:



**sudo adduser andres profesores**

**Nota:** este comando también sirve para dar de alta (crear) usuarios en el sistema. Esta utilidad se verá más adelante (parecido a *useradd*).

### 1.3.6. Quitar usuarios de un grupo

**deluser.** Permite eliminar un usuario de un grupo que debe existir previamente. La sintaxis es:

**deluser nombre-usuario nombre-grupo**

Ejemplo:

Eliminamos el usuario “andres” del grupo “profesores”:

**sudo deluser andres profesores**

**Nota:** este comando también sirve para eliminar usuarios del sistema (parecido a *userdel*). Esta utilidad se verá más adelante.

### 1.3.7. Otros comandos de grupos

**groups.** Lista los grupos a los que pertenece el usuario. La sintaxis es:

**groups nombre-usuario**

Ejemplo:

Listamos los grupos a los que pertenece el usuario andres:

**groups andres**

Ejemplo:

El usuario activo lista los grupos a los que pertenece:

**Groups**

## 1.4. Gestión de permisos sobre directorios y ficheros

### 1.4.1. Modificación de los Permisos de un Fichero

**chmod.** Permite modificar los permisos de uno o varios ficheros. Su sintaxis es:

**chmod mascara fichero/s**

donde la **máscara** puede ser *octal* o *simbólica*.

Ejemplo:

**chmod 652 practica**

Cada cifra de la máscara representa los valores para lectura, escritura y ejecución (rwx)

**Nota:** escribir **ls -la** y mirar la primera columna de los datos representados

```
drwxr-xr-x  5 ttlaoc ttlaoc  4096 2014-03-26 20:15 .
drwxr-xr-x 31 ttlaoc ttlaoc  4096 2014-03-26 20:18 ..
-rwxrwxrwx  1 ttlaoc ttlaoc  1538 2012-02-29 23:23 examen.sh
-rw-rw-rw-  1 ttlaoc ttlaoc  1530 2012-02-29 23:22 examen.sh~
drwxrwxrwx  2 ttlaoc ttlaoc  4096 2014-03-26 20:20 hola
drwxr-xr-x 23 ttlaoc ttlaoc  4096 2012-01-09 12:28 Prog web
-rw-r--r--  1 ttlaoc ttlaoc 477863 2012-01-09 13:11 Prog web_01.zip
-rwxrwxrwx  1 ttlaoc ttlaoc   182 2012-04-19 21:10 prueba.sh
-rw-rw-rw-  1 ttlaoc ttlaoc   184 2012-04-19 21:09 prueba.sh~
drwxr-xr-x  6 ttlaoc ttlaoc  4096 2014-03-24 13:14 Scripts
-rw-r--r--  1 ttlaoc ttlaoc    60 2014-03-21 19:58 test
```

En la fila marcada, si nos fijamos en la primera columna de datos (ignoramos la primera “d”), nos encontramos con: rwxrwxrwx. Se puede comprobar que se repite tres veces la secuencia “rwx”: la “r” de “read” (lectura), la “w” de “write” (escritura) y la “x” de “execute” (ejecución). Con cada letra se especifica qué tipo de permiso se está otorgando (permiso de lectura, escritura y/o ejecución).

La primera secuencia “rwx” hace referencia a los permisos del propietario del fichero, la segunda secuencia “rwx” hace referencia a los permisos del grupo al que se asigna el fichero y la tercera secuencia “rwx” hace referencia a otros usuarios (usuarios que no son el propietario ni los usuarios que pertenecen al grupo al que

se ha asignado el fichero).

De este modo, el archivo “hola” (el marcado), permite lectura, escritura y ejecución a todos los usuarios del sistema. Sin embargo, el archivo “Prog web” (el siguiente al marcado), da todos los permisos al propietario del fichero y lectura y ejecución al resto de los usuarios del sistema (rwxr-xr-x).

Para representar las tres secuencias “rwx”, se puede usar notación octal de una forma sencilla, donde dar permiso es un 1 y la ausencia de permiso es 0. De este modo, un ejemplo donde los permisos sean rwxr-xr--, quedaría en octal: 754.

El propietario de “practica” tendrá permiso de lectura y escritura (4+2=6), el grupo de lectura y ejecución (4+1=5) y otros de escritura (2).

Si quisiéramos expresar lo mismo mediante una máscara simbólica, pondríamos:

**chmod u=rw, g=rx, o=w practica**

La máscara simbólica está formada por tres códigos:

**a) Clases de usuarios:**

<u>Símbolo</u>	<u>Significado</u>
u	propietario
g	grupo
o	otros
a	todos (propietario, grupo y otros)

**b) Permisos:**

<u>Símbolo</u>	<u>Significado</u>
r	lectura
w	escritura
x	ejecución

**c) Operación:**

<u>Símbolo</u>	<u>Significado</u>
+	añadir permisos
-	quitar permisos
=	asignar permisos

Ejemplo:

**ls -l**

Aparece por pantalla:

```
-rw-r-x-w-    1 cbt staff 512 Feb 09    18:55 practica
```

**chmod u+x, g-x practica**

**ls -l**

Aparece por pantalla:

```
-rwxr---w-    1 cbt staff 512 Feb 09    18:55 practica
```

Da permiso de ejecución del fichero al propietario y le quita el permiso de ejecución al grupo.

Ejemplos:

```
chmod +r arch.txt      # agrega permisos de lectura a todos los
                        # usuarios(solo a algunos elegidos previamente)
chmod u+w arch.txt     # agrega permisos de escritura al dueño
chmod -x arch.txt      # elimina el permiso de ejecución a todos
                        # los usuarios
chmod u=rw,go= arch.txt # establece los permisos de lectura y escritura
                        # al dueño y elimina todos los permisos a
                        # los demás usuarios
```

#### 1.4.2. Cambio del Propietario de un Fichero

**chown.** Este comando cambia el propietario de un fichero. La sintaxis es:

**chown nuevo\_propietario fichero/s**

Donde:

**nuevo\_propietario** se sustituye por el nombre o el identificador de usuario del nuevo propietario.

Ejemplo:

**chown pepe carta**

El usuario “pepe” pasa a ser el propietario del fichero “carta”.

### 1.4.3. Cambio de Grupo a un Fichero

**chgrp.** Este comando cambia de grupo a un fichero. La sintaxis es:

**chgrp nuevo\_grupo fichero/s**

Donde:

**nuevo\_grupo** se sustituye por el nombre o el identificador de grupo del nuevo grupo.

Ejemplo:

**chgrp usuarios1 carta1 carta2**

El grupo de los ficheros “carta1” y “carta2” pasa a ser el grupo “usuarios1”