

# Linux

## 1. ¿Cómo saber a qué grupos pertenece un usuario?

En sistemas operativos Linux, los usuarios pueden pertenecer a más de un grupo.

Veamos dos formas para saber a que grupos pertenece un usuario:

### 1) Comando groups:

Ejecutando el comando groups, nos aparecerán los grupos a los que pertenece el usuario actual.

Si ejecutamos groups seguido de un nombre de usuario, veremos los grupos a los que pertenece el usuario indicado.

Ejemplo:

```
[root@LINUX1 ~]# groups root
root : root bin daemon sys adm disk wheel
```

### 2) Comando id:

El comando id, nos muestra información acerca de la identidad del usuario.

Si ejecutamos el comando id, con los parámetros -Gn, conseguiremos obtener la lista de grupos a los que pertenece el usuario.

```
[root@LINUX1 ~]# id -Gn root
root bin daemon sys adm disk wheel
```

El parámetro -G, muestra todos los grupos a los que pertenece , mientras que el parámetro -n muestra el nombre en vez del número de grupo.

## 2. Comando chown

Por norma general, la persona que crea un archivo o directorio permanece como propietaria de este.

No obstante, si hay que cambiar al usuario o grupo propietario en un archivo o directorio, Linux pone a tu disposición “chown”. Este comando debe su nombre a los términos en inglés “**change owner**”, esto es, cambiar propietario.

Para asignar un nuevo propietario a un archivo o directorio, hay que indicar el nuevo propietario en el comando introduciendo el **nombre de la persona o su número de identificación**. La información relevante a la que accede el comando `chown` se encuentra en `/etc/passwd` y `/etc/group`.

El comando `chown` siempre sigue la misma sintaxis básica:

```
$ chown [Opciones] [Usuario] [:[Grupo]] Archivo
```

No obstante, hay varias formas de combinar el nombre del nuevo propietario y el grupo. El comando `chown` permite lo siguiente:

- **nombre\_usuario:nombre\_grupo**: almacena usuario y grupo al mismo tiempo.
- **:nombre\_grupo**: solo se cambia la propiedad del grupo; no el usuario.
- **nombre\_usuario**: solo se cambia al usuario.
- **nombre\_usuario:**: se cambia al usuario; el grupo es el grupo por defecto del usuario registrado.

Hay varias opciones con las que puedes personalizar el modo de funcionar del comando. Estas son las más importantes:

- **-c o -changes**: con esta opción solo se muestran los archivos en los que se ha cambiado el propietario.
- **-f o -force**: si utilizas esta opción, se ocultan los mensajes de error sobre intentos de cambio fallidos.
- **-R o -recursive**: con esta opción puedes asegurarte de que la ejecución del comando `chown` también se aplica a los subdirectorios.
- **-v o -verbose**: con esta opción, puedes ver lo que el comando `chown` está haciendo en un momento concreto.

Descubre algunos ejemplos de cómo se usa el comando `chown` de Linux para que te puedas hacer una idea precisa:

```
$ chown Ana archivodeejemplo.txt
```

Con este comando, Ana se convierte en la nueva propietaria del archivo archivodeejemplo.txt.

```
$ chown $USER archivodeejemplo.txt
```

El usuario actual se convierte en propietario del archivo archivodeejemplo.txt.

```
$ chown -cR Ana:Datos /media/documents
```

Ana se convierte en propietaria del directorio /media/documents y de todos los subdirectorios asociados. Además, los directorios y las subcarpetas se transfieren al grupo “Datos”.

### 3. Comando **chgrp**

El comando ‘chgrp’ en Linux se utiliza para cambiar el grupo de un archivo o directorio. La sintaxis básica de este comando es la siguiente:

**chgrp** [opciones] grupo archivo/directorio

Donde "grupo" es el nombre del grupo al que se desea cambiar y "archivo/directorio" es el archivo o directorio al que se le desea cambiar el grupo. Es importante destacar que solo el usuario root o el propietario del archivo o directorio pueden cambiar el grupo.

El comando ‘chgrp’ en Linux también cuenta con opciones avanzadas que permiten realizar cambios más específicos en los grupos. Algunas de las opciones más comunes incluyen:

‘-R’: Cambiar recursivamente el grupo de un directorio y todos sus subdirectorios y archivos.

‘-h’: Cambiar el grupo de los enlaces simbólicos en lugar del archivo o directorio al que están enlazados.

‘-from’: Cambiar el grupo solo si ya pertenece a un grupo específico.

‘-reference’: Cambiar el grupo de un archivo o directorio basado en las propiedades de otro archivo o directorio.

Estas opciones permiten un mayor control sobre los cambios de grupo y pueden ser útiles en situaciones más complejas.

De cualquiera de las siguientes maneras, se puede cambiar la propiedad del grupo de un archivo:

```
chown :david misnotas
```

```
chgrp david misnotas
```

### 4. Comando **newgrp**

El comando **newgrp** permite a un usuario anular el grupo primario actual. Puede ser útil cuando trabajamos en un directorio donde todos los archivos deben tener la misma propiedad de grupo. Supongamos que tenemos un directorio llamado compartir en su servidor de intranet donde diferentes equipos almacenan fotos de marketing. El grupo es «share». A medida que diferentes usuarios colocan archivos en el directorio, los grupos principales de los archivos pueden confundirse. Cada vez que se agregan nuevos archivos, podemos ejecutar **chgrp** para corregir cualquier confusión configurando el grupo para compartir.

Esto cambia su grupo primario para compartir, por lo que todos los archivos que colocan en el directorio tendrán automáticamente el grupo compartido, en lugar del grupo primario del usuario. Una vez que hayan terminado, los usuarios pueden volver a su grupo principal habitual con (por ejemplo):

```
newgrp arnau
```

*(Ver también en el pdf completo Usuarios y Grupos. Permisos)*