



Cambiar los permisos de los archivos

El comando `chmod` se utiliza para cambiar los permisos de un archivo o directorio. Sólo el usuario raíz o el usuario propietario del archivo puede cambiar los permisos de un archivo.

Considere esto

¿Por qué el comando se llama `chmod` en lugar de `chperm` ? Los permisos solían denominarse *modos de acceso*, por lo que el comando `chmod` realmente significa *cambiar los modos de acceso* (**change access mode**).

Hay dos métodos para cambiar permisos usando el comando `chmod` : el método *simbólico* y el método *octal*. El método simbólico es útil para cambiar un conjunto de permisos a la misma vez. El método octal o numérico requiere conocer el valor octal de cada uno de los permisos y requiere que los tres conjuntos de permisos (usuario, grupo, otros) se especifiquen cada vez. Para simplificar las cosas, solamente trataremos el método simbólico. Para obtener más información sobre el método octal eche un vistazo al curso [NDG Linux Essentials](#).

Siga leyendo

Utilice el siguiente comando para cambiar al directorio `Documents` :

```
sysadmin@localhost:~$ cd ~/Documents
```

El método simbólico

```
chmod [<COJUNTO DE PERMISOS><ACCIÓN><PERMISOS>]... ARCHIVO
```

Para usar el método simbólico de `chmod` primero debe indicar qué conjunto de permisos se está cambiando:

```
chmod [ <CONJUNTO DE PERMISOS> <ACCIÓN><PERMISOS>]... ARCHIVO
```

Símbolo	Significado
u	Usuario: El usuario propietario del archivo.
g	Grupo: El grupo propietario del archivo.
o	Otros: Cualquier otro que no sea el usuario propietario o un miembro del grupo propietario.
a	Todos: Se refiere al usuario, grupo, y todos los demás.

A continuación, especifique un símbolo para la acción:

```
chmod [<CONJUNTO DE PERMISOS> <ACCIÓN> <PERMISOS>] ... ARCHIVO
```

Símbolo	Significado
+	Añadir permiso, si es necesario
=	Especificar el permiso exacto
-	Eliminar el permiso, si es necesario

Después del símbolo de acción, especifique uno o más permisos.

```
chmod [<CONJUNTO DE PERMISOS><ACCIÓN> <PERMISOS> ] ... ARCHIVO
```

Símbolo	Significado
r	leer (<i>read</i>)
w	escribir (<i>write</i>)
x	ejecutar (<i>execute</i>)

Finalmente, añada un espacio y los nombres de ruta para los archivos a los que quiere asignar los permisos.

```
chmod [<CONJUNTO DE PERMISOS><ACCIÓN><PERMISOS>] ... ARCHIVO
```

El archivo `hello.sh` utilizado en los ejemplos de la página anterior es un script. Un script es un archivo que se puede ejecutar, similar a un comando:

```
sysadmin@localhost:~/Documents$ ls -l hello.sh
-rw-r--r-- 1 sysadmin sysadmin 647 Dec 20 2017 hello.sh
```

Sin embargo, actualmente, el permiso de ejecución no ha sido establecido para ninguno de los grupos de permisos:

```
-rw- r- r- 1 sysadmin sysadmin 647 Dec 20 2017 hello.sh
```

De esta manera, intentar ejecutar este script utilizando la siguiente sintaxis produce un error:

```
sysadmin@localhost:~/Documents$ ./hello.sh
-bash: ./hello.sh: Permission denied
```

Dado que el sistema está actualmente conectado como usuario `sysadmin`, y `sysadmin` es el propietario de este archivo, otorgar el permiso de ejecución al usuario propietario debería permitirle ejecutar este script. Usando el comando `chmod` con el carácter `u` para representar el conjunto de permisos del usuario propietario, y agregando el carácter `+` para indicar que se añade un permiso y el carácter `x` para representar el permiso de ejecución, el comando deberá ejecutarse con la siguiente sintaxis:

```
sysadmin@localhost:~/Documents$ chmod u+x hello.sh
```


No obtener un resultado/mensaje indica que el comando se ha realizado correctamente. Confírmelo examinando los permisos con el comando `ls -l`:

```
sysadmin@localhost:~/Documents$ ls -l hello.sh
-rwxr--r-- 1 sysadmin sysadmin 647 Dec 20  2017 hello.sh
```

El usuario propietario ahora posee permiso para ejecutar:

```
-rwxr--r-- 1 sysadmin sysadmin 647 Dec 20  2017 hello.sh
```

Finalmente, intente ejecutar el script de nuevo. Utilice la sintaxis de comando que se muestra a continuación:

```
./hello.sh
```

```
sysadmin@localhost:~/Documents$ ./hello.sh
```

```
( Hello World! )
```

```
-----
```

```
\
```

```
\
```

```
<(^)
```

```
( )
```

A tener en cuenta

Observe que para ejecutar el script en el ejemplo anterior, la siguiente combinación de caracteres `./` se colocó antes del nombre del script.

```
./hello.sh
```

Esto indica que el “comando” debe ejecutarse desde el directorio actual.