

## Permisos de archivos en Linux y Windows

### Permisos en Linux: El lenguaje de los bits

En Linux, cada archivo tiene un conjunto de permisos que se aplican a tres tipos de usuarios:

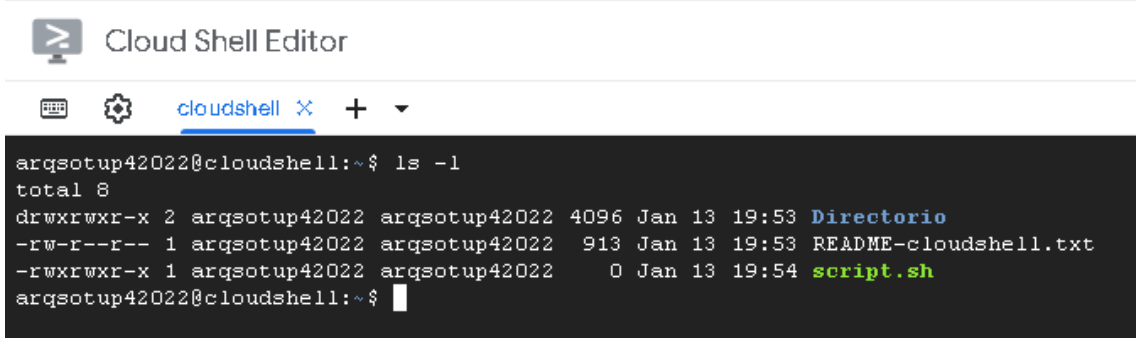
- **Propietario:** La persona dueña del archivo.
- **Grupo:** Un conjunto de usuarios con acceso compartido al archivo.
- **Otros:** Cualquier otro usuario del sistema.

Para cada tipo de usuario, hay tres permisos básicos:

- **Lectura (r):** Permite ver el contenido del archivo, como leer un documento o ver una imagen.
- **Escritura (w):** Permite modificar el contenido del archivo, como editar un documento o cambiar una imagen.
- **Ejecución (x):** Permite ejecutar el archivo como un programa.

### Visualizando los permisos: ls -l

El comando `ls -l` lista los archivos y directorios junto con sus permisos. La salida se ve así:



```
Cloud Shell Editor

argsotup42022@cloudshell:~$ ls -l
total 8
drwxrwxr-x 2 argsotup42022 argsotup42022 4096 Jan 13 19:53 Directorio
-rw-r--r-- 1 argsotup42022 argsotup42022 913 Jan 13 19:53 README-cloudshell.txt
-rwxrwxr-x 1 argsotup42022 argsotup42022 0 Jan 13 19:54 script.sh
argsotup42022@cloudshell:~$
```

- El primer carácter indica el tipo de archivo (- para archivo normal, d para directorio, l para enlace simbólico).
- Los siguientes nueve caracteres representan los permisos, divididos en tres grupos de tres: propietario, grupo y otros.
- Cada grupo de tres caracteres indica los permisos de lectura (r), escritura (w) y ejecución (x) para esa categoría de usuario.

Por ejemplo, `-rw-r--r--` significa que el propietario tiene permisos de lectura y escritura, mientras que su grupo y otros usuarios solo tienen permiso de lectura.

Acá entran en escena los bits. Un bit es la unidad mínima de información en una computadora, y puede tener dos valores: 0 o 1. Linux usa 3 bits para representar cada permiso (lectura, escritura y ejecución), uno para cada tipo de usuario.

Esto puede observarse en la siguiente tabla:

Permiso	Bit	Valor numérico
Sin permiso	000	0
Ejecución (x)	001	1
Escritura (w)	010	2
Escritura (w) + Ejecución (x)	011	3
Lectura (r)	100	4
Lectura (r) + Ejecución (x)	101	5
Lectura (r) + Escritura (w)	110	6
Lectura (r) + Escritura (w) + Ejecución (x)	111	7

### Modificando los permisos: chmod

El comando chmod permite modificar los permisos de archivos y directorios.

Se puede usar de dos maneras:

#### 1. Notación numérica:

Cada permiso tiene un valor numérico (ver valores de tabla):

- r (lectura) = 4
- w (escritura) = 2
- x (ejecución) = 1

Sumando los valores de los permisos que se desean otorgar a cada categoría de usuario, se obtiene un número de tres dígitos que representa los permisos del archivo

#### 2. Notación simbólica:

Se utilizan letras para representar las categorías de usuarios (u, g, o, a para todos) y símbolos para agregar (+), quitar (-) o asignar (=) permisos (r, w, x). Por ejemplo:

- chmod o+r archivo.txt agrega permiso de lectura para otros.
- chmod g-w archivo.txt quita el permiso de escritura al grupo.
- chmod u=rwx archivo.txt asigna al propietario permisos de lectura, escritura y ejecución, eliminando cualquier permiso anterior que tuviera.

#### Opciones adicionales:

- -R: Aplica los permisos recursivamente a un directorio y su contenido.
- --reference=archivo\_referencia: Copia los permisos de un archivo a otro.

### Chown (Change Owner)

Este comando cambia la propiedad de un archivo o directorio, definiendo quién tiene el control principal sobre ese archivo/directorio, pudiendo decidir quién más puede acceder a él y qué permisos tienen (leer, escribir, ejecutar).

Por ejemplo, si "usuario1" crea un archivo, inicialmente él es el dueño. Pero si queremos que "usuario2" sea el nuevo dueño, usamos el comando:

```
chown usuario2 archivo.txt
```

### Asignando permisos

#### Ejemplo 1: Números para permisos

Si quieres dar todos los permisos al dueño del archivo, sumas  $4 + 2 + 1 = 7$ . Si quieres que el grupo solo pueda leer, le das 4. Y si quieres que otros solo puedan leer, también les das 4.

Entonces, `chmod 744 archivo.txt` significa:

- Dueño: Todos los permisos (7)
- Grupo: Solo lectura (4)
- Otros: Solo lectura (4)

Puedes combinar estos números para dar diferentes permisos. Por ejemplo, `chmod 755` da permisos de lectura y ejecución al grupo y a otros.

#### Ejemplo 2: Permisos en cascada

Si tienes una carpeta con muchos archivos y subcarpetas dentro, y quieres que todos tengan los mismos permisos, usas la opción `-R` (recursivo). Es como una cascada: los permisos caen desde la carpeta principal hacia abajo.

Por ejemplo, `chmod 755 -R carpeta` le da permisos de lectura y ejecución al grupo y a otros, tanto a la carpeta "carpeta" como a todo lo que tiene dentro.

#### Ejemplo 3: Permisos con letras

En vez de usar números, también puedes usar letras para cambiar los permisos:

- u: Dueño del archivo
- g: Grupo
- o: Otros
- a: Todos (dueño, grupo y otros)

También usas símbolos para decir si quieres agregar (+), quitar (-) o asignar (=) permisos:

- r: Leer
- w: Escribir
- x: Ejecutar

Por ejemplo, `chmod g+r archivo.txt` agrega permiso de lectura al grupo.

#### Ejemplo 4: Solo lectura para otros

`chmod o=r archivo.txt` significa: "quiero que 'otros' solo puedan leer este archivo, y quitarles cualquier otro permiso que tuvieran antes".

#### Ejemplo 5: Permiso para ejecutar

`chmod ug+x archivo.txt` agrega el permiso de ejecución (x) al dueño (u) y al grupo (g). Ahora ambos podrán ejecutar el archivo si es un programa.

#### Ejemplo 6: Permisos personalizados

`chmod u=rwx,g=rw,o=r archivo.txt` es una forma de dar permisos específicos a cada uno:

- Dueño: Todos los permisos (leer, escribir, ejecutar)
- Grupo: Leer y escribir
- Otros: Solo leer

#### Ejemplo 7: Quitar todos los permisos a otros

`chmod o= archivo.txt` quita todos los permisos a "otros". Es como decir "nadie más que el dueño y el grupo puede hacer nada con este archivo".

#### Ejemplo 8: Copiar permisos

Si tienes un archivo con los permisos que quieres, y quieres que otro archivo tenga los mismos permisos, puedes usar la opción `--reference`. Es como copiar y pegar los permisos.

Por ejemplo, `chmod --reference=archivo1.txt archivo2.txt` hace que `archivo2.txt` tenga los mismos permisos que `archivo1.txt`.

#### Ejemplo 9: Ejecutar solo carpetas

Si tienes una carpeta con archivos y subcarpetas, y solo quieres que se puedan ejecutar las carpetas (pero no los archivos), usas `chmod a+X *`. La X mayúscula solo se aplica a carpetas.

## Permisos en Windows: Control de acceso granular

A diferencia de Linux, que utiliza un sistema de permisos basado en propietario, grupo y otros, Windows ofrece un control más granular a través de las **Listas de Control de Acceso (ACL)**.

### ¿Qué es una ACL?

Consideremos una lista detallada donde se especifica quién tiene acceso a un archivo y qué puede hacer con él. Cada entrada en esta lista, o ACL, contiene:

- **Usuario o grupo:** Indica a quién se le otorgan los permisos. Puede ser un usuario individual o un grupo de usuarios.
- **Permisos:** Define las acciones permitidas para ese usuario o grupo, como lectura, escritura, modificación, ejecución, etc.

### ¿Cómo se gestionan las ACLs?

Windows ofrece dos formas principales de gestionar las ACLs:

- **CLI (Interfaz de Línea de Comandos):** Utilizando el comando `icacls`, puedes ver, modificar y configurar las ACLs desde la consola. Esto es útil para automatizar tareas o realizar cambios masivos.
- **GUI (Interfaz Gráfica de Usuario):** A través del Explorador de Windows, puedes acceder a las propiedades de seguridad de un archivo y modificar las ACLs de forma visual. Esta opción es más intuitiva para usuarios no familiarizados con la línea de comandos.

### Permisos más comunes

- **Control total:** Permite realizar cualquier acción con el archivo o carpeta, incluyendo modificarlo, eliminarlo y cambiar sus permisos.
- **Modificar:** Permite leer, escribir, ejecutar y eliminar el archivo o carpeta.
- **Leer y ejecutar:** Permite ver el contenido y ejecutar el archivo, pero no modificarlo.
- **Leer:** Permite solo ver el contenido del archivo o carpeta.
- **Escribir:** Permite modificar el archivo, pero no leerlo ni ejecutarlo.

### ¿Cómo se gestionan los permisos?

Para ver o modificar los permisos de un archivo o carpeta:

1. Haz clic derecho en el archivo o carpeta y selecciona "**Propiedades**".
2. Ve a la pestaña "**Seguridad**".
3. Verás una lista de usuarios y grupos con sus permisos.
4. Para cambiar los permisos, selecciona un usuario o grupo y haz clic en "**Editar**".
5. Puedes agregar o quitar usuarios o grupos, y asignar o quitar permisos específicos.

#### Consejos para gestionar los permisos:

- **Principio de mínimo privilegio:** Otorga solo los permisos necesarios a cada usuario o grupo.
- **Usa grupos:** En lugar de asignar permisos a usuarios individuales, crea grupos y asigna los permisos a los grupos. Esto facilita la administración.
- **Revisa los permisos heredados:** Las carpetas heredan los permisos de sus carpetas padre. Asegúrate de que los permisos heredados sean apropiados.
- **Ten cuidado al otorgar "Control total":** Este permiso otorga un control absoluto sobre el archivo o carpeta, incluyendo la capacidad de cambiar los permisos y eliminar el archivo.

#### Comparación final: Linux vs Windows

- **Similitudes:** Tanto Windows como Linux emplean los permisos básicos de lectura, escritura y ejecución para controlar el acceso a los archivos. Estos permisos determinan si un usuario puede ver el contenido del archivo, modificarlo o ejecutarlo como un programa.
- **Interfaces:** Ambos sistemas operativos ofrecen la flexibilidad de gestionar los permisos a través de la línea de comandos (CLI) o mediante una interfaz gráfica de usuario (GUI). Esto permite a los usuarios elegir la forma que mejor se adapte a sus necesidades y preferencias.
- **Sistemas de control:** Linux utiliza un sistema de permisos basado en bits, donde se asignan permisos al propietario del archivo, al grupo asociado y a otros usuarios. En contraste, Windows emplea Listas de Control de Acceso (ACL), que permiten un control más granular al especificar permisos para usuarios y grupos específicos.
- **Asignación de permisos:** En Linux, los permisos se definen para tres categorías generales: propietario, grupo y otros. Windows, por otro lado, permite una asignación más precisa de permisos, ya que se pueden configurar para usuarios y grupos específicos, ofreciendo un mayor control sobre el acceso a los archivos.