Permisos en Linux - EXT4

Permisos (I)

- En el sistema de ficheros ext4 se especifican los permisos de un archivo o directorio:
  - Del usuario propietario. Por defecto es el creador.
  - Del grupo propietario. Por defecto el principal del creador.
  - Del resto de usuarios
- Recuerda que en linux, un archivo es todo, incluyendo los dispositivos.
- **root** es un usuario que puede acceder a cualquier recurso, independientemente de los permisos.

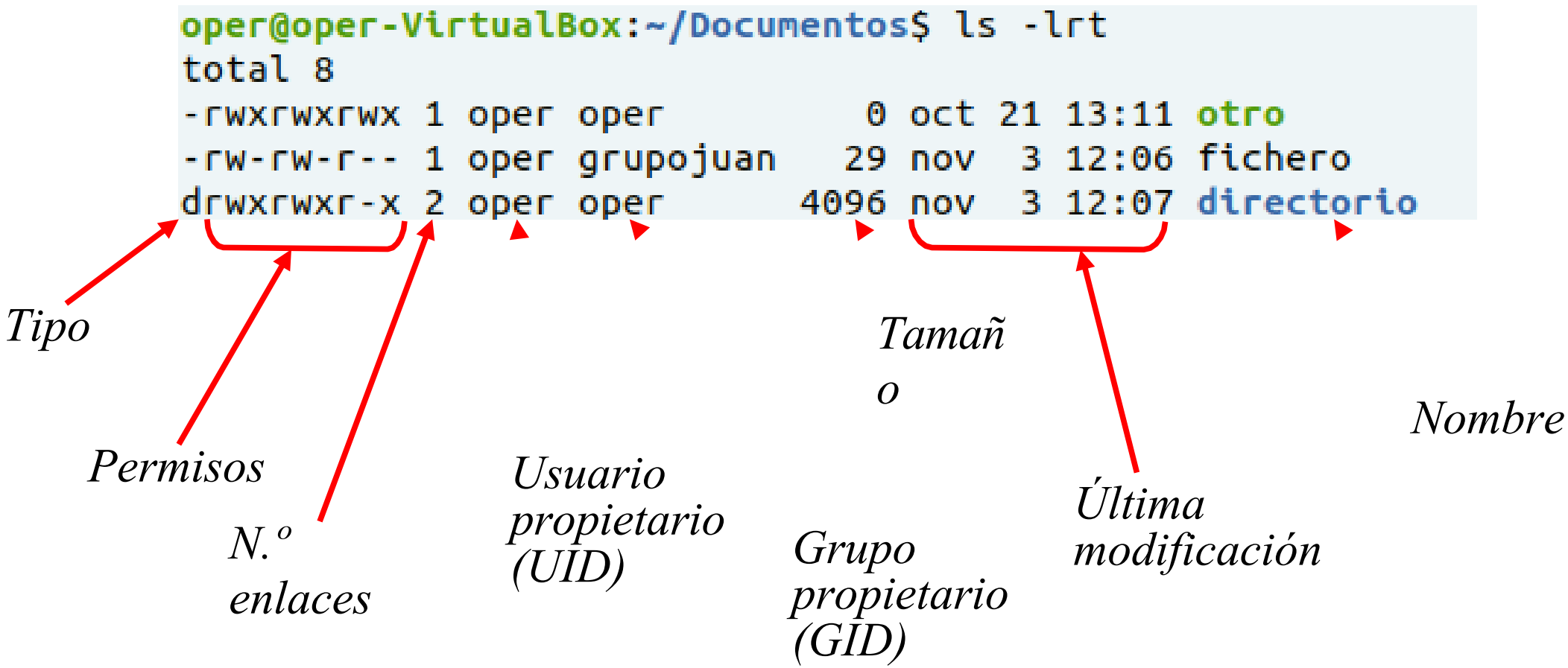
Permisos (II)

- El inodo guarda metadatos del archivo, así como los identificadores de los bloques en dónde se almacena el contenido.
- El UID del archivo (usuario propietario) y GID (grupo propietario) se almacena en el inodo, junto con los permisos.
- Si ejecutas un *ls -l* (formato extendido), verás estos metadatos

```
oper@oper-VirtualBox:~/Documentos$ ls -lrt
total 8
-rwxrwxrwx 1 oper oper          0 oct 21 13:11 otro
-rw-rw-r-- 1 oper grupojuan    29 nov  3 12:06 fichero
drwxrwxr-x 2 oper oper       4096 nov  3 12:07 directorio
```

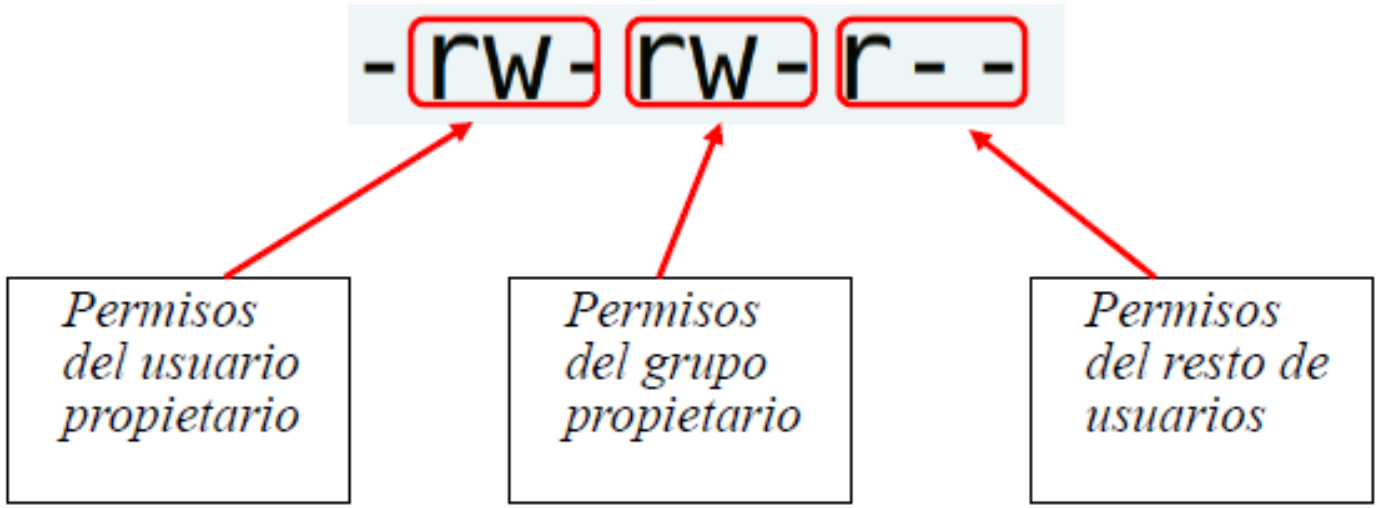
Permisos (III)

```
oper@oper-VirtualBox:~/Documentos$ ls -lrt
total 8
-rwxrwxrwx 1 oper oper          0 oct 21 13:11 otro
-rw-rw-r-- 1 oper grupojuan    29 nov  3 12:06 fichero
drwxrwxr-x 2 oper oper       4096 nov  3 12:07 directorio
```



Permisos – bits (I)

- Los permisos son un conjunto de 9 bits, agrupados en de tres en tres:



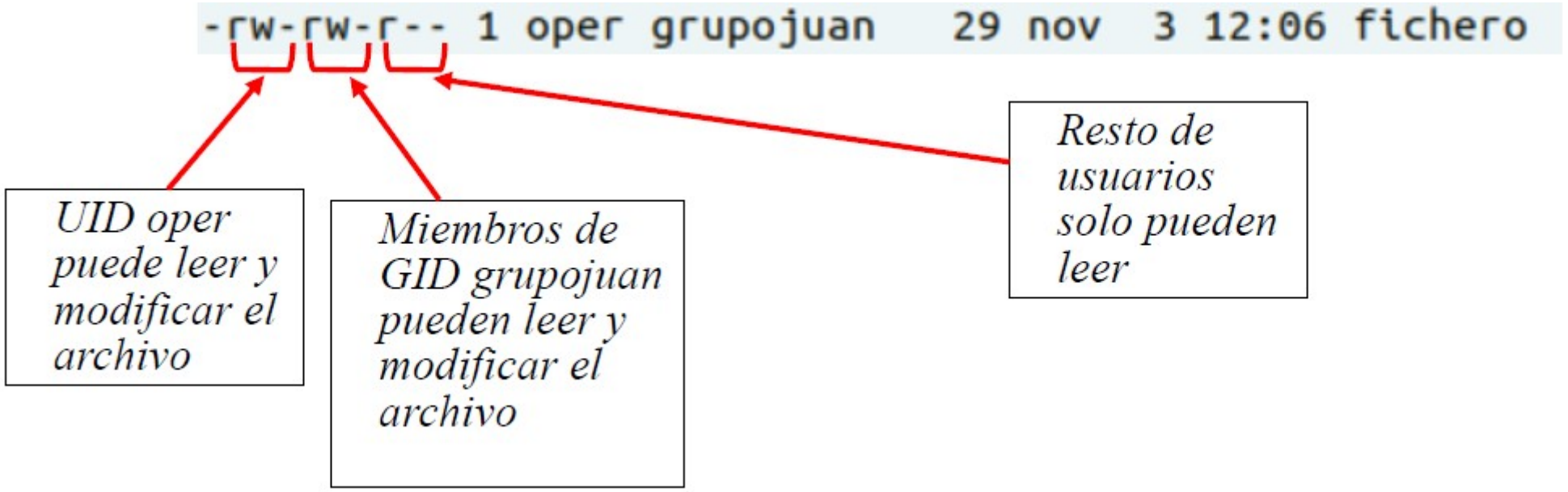
Permisos – bits (II)

- Cada grupo de 3 bits indican:
  - Primer bit (r): permiso para leer.
  - Segundo bit (w): permiso para escribir.
  - Tercer bit (x): permiso de ejecución.

Permiso	Aplicado a archivo	Aplicado a directorio
r	Visualizar contenido	Listar contenido
w	Modificar contenido, así como permiso, usuario y grupo propietario	Crear y borrar dentro del directorio
x	Ejecutar	Entrar en el directorio (cd)

Permisos – bits (III)

- Si aparece una letra es que el bit está activo, y se dispone del permiso. Si aparece "-" significa que no se dispone del permiso.



Permisos – permisos en directorios

- Para poder listar y acceder a directorios debemos tener permisos **r** y **x**.
- Además, aunque tengamos permiso de acceso a un archivo, si no tenemos acceso a cualquiera de los directorios de la ruta absoluta, no podremos acceder.
  - Si un usuario tiene archivos en una carpeta a la que solo el o ella deba acceder, puede dejar permisos rwx exclusivos a propietario, y --- a grupo y resto.

```
drwx------ 2 oper oper      4096 nov  3 12:07 directorio
```

  - De este modo, no se tiene que preocupar de los permisos dentro del directorio.

ACLs en Linux

- Como ves, las posibilidades de gestión de permisos en Linux son más limitadas que en Windows.
- Sin embargo, es posible emplear ACLs también en Linux. Los sistemas de ficheros ext4 y btrfs lo permiten por defecto.
- Nosotros no trabajaremos con ellas.
  - *getfacl: mostrar ACL*
  - *setfacl: modificar ACL*

Bits especiales: set-uid

- Cuando un usuario ejecuta un archivo, el proceso se ejecuta con permisos de dicho usuario, independientemente de quien sea el propietario.
- Si este bit está activo, entonces el proceso se ejecuta con los privilegios del propietario, no del ejecutor.

```
-rwsr-xr-x 1 root root 67816 jul 21  2020 /bin/su
-rwsr-xr-x 1 root root 68208 jul 15 00:08 /bin/passwd
```

- Se identifica por tener en el tercer bit de permisos del propietario una **s** en lugar de una **x**.

Bits especiales: set-gid

- Se identifica porque en el tercer bit del grupo aparece una **s** en lugar de una **x**.
- Se comporta diferente según sea un fichero o un directorio:
  - Si es un fichero ejecutable → equivalente al set-uid, pero se ejecuta tomando como grupo el del propietario del archivo, no el del ejecutor.
  - Si es un directorio → los archivos y subdirectorios creados dentro tendrán como grupo propietario el mismo que el del directorio, no el del ejecutor.

Bits especiales: sticky bit

- Sólo tiene sentido sobre directorios. Se identifica por tener en el tercer bit de los permisos del resto de usuarios una **t** en lugar de una **x**.
- En un directorio con sticky bit activo, sólo el propietario de un archivo puede borrarlo. El resto de usuarios no podrán, aunque a nivel de permisos del directorio se permita.
- Ejemplo más típico: directorios de archivos temporales */tmp o /var/tmp*, en los que cualquier usuario puede crear archivos, pero solo el propietario debe poder borrarlos.

```
drwxrwxrwt  19 root root      4096 nov  3 17:10 tmp
```

- También usado, por ejemplo, en FTP anónimos, para que solo borre cada archivo su propietario.

Modificación de permisos

- Para modificar los permisos de archivos o directorios se emplea:

  chmod [opciones] permisos archivos/directorios

- La opción más usada es “-R”, para aplicar el cambio de forma recursiva.
- Los permisos se pueden especificar de dos modos alternativos:
  - Octal
  - Carácter

Modificación de permisos – Octal (I)

- Los permisos se expresan mediante 3 números en base 8.
- Cada grupo de 3 bits se transforma en un número de 0 a 7.
- Si bit activo, es un 1. Si no está activo (“-”), es un 0.
- Para cada uno de los 3 grupos, convertimos primero en binario y luego a octal
  - rw- $\rightarrow 110)_2 \rightarrow 6)_8$
- Ejemplos:
  - rw-r-----: 640
  - rwxrwxrwx: 777
  - rw-rw-r--: 664

```
oper@oper-VirtualBox:~/Documentos/directorio$ ls -l fichero
-rw-rw-r-- 1 oper oper 0 nov  3 17:30 fichero
oper@oper-VirtualBox:~/Documentos/directorio$ chmod 754 fichero
oper@oper-VirtualBox:~/Documentos/directorio$ ls -l fichero
-rwxr-xr-- 1 oper oper 0 nov  3 17:30 fichero
```

Modificación de permisos – Octal (II)

- En caso de querer usar los bits especiales, debemos sumar al cálculo anterior:
  - sticky bit: sumamos 1000
  - set-gid: sumamos 2000
  - set-uid: sumamos 4000
- Ejemplos:

  1775: rwxrwxr-t

  2656: rw-r-srw-

  4764: rwsrw-r--

```
oper@oper-VirtualBox:~/Documentos/directorio$ ls -lrt
total 4
drwxrwxr-x 2 oper oper 4096 nov  4 10:07 dir_temp
-rw------- 1 oper oper    0 nov  4 10:08 fichero
oper@oper-VirtualBox:~/Documentos/directorio$ chmod 1775 dir_temp/
oper@oper-VirtualBox:~/Documentos/directorio$ chmod 4764 fichero
oper@oper-VirtualBox:~/Documentos/directorio$ ls -lrt
total 4
drwxrwxr-t 2 oper oper 4096 nov  4 10:07 dir_temp
-rwsrw-r-- 1 oper oper    0 nov  4 10:08 fichero
```

Modificación de permisos – Carácter

- En este caso la modificación se expresa mediante una combinación de caracteres.

[u] [g] [o] [a] {+|=|-} [r] [w] [x] [s] [t]

u → usuario propietario *g → grupo propietario* *o → resto de usuarios* *a → todos los usuarios*	*+ → se añade el permiso al valor actual* *- → se elimina el permiso del valor actual* *= → resto de usuarios*	*r → lectura* *w → escritura* *x → ejecución* *s → set-uid o set-gid* *t → sticky bit*

- Se pueden incluir varias combinaciones, separadas por “,”.
- Si no se especifica a quién aplica el cambio, por defecto es “a”.

Modificación de permisos – Carácter- Ejemplos

chmod u+r fich	Añadimos permiso de lectura a usuario propietario
chmod u+r-w fich	Añadimos permiso de lectura a usuario propietario y le quitamos el de escritura
chmod ug+r fich	Añadimos permiso de lectura a usuario y grupo propietario
chmod +r fich	Añadimos el permiso de lectura para todos (a)
chmod u+w,o-r fich	Al usuario propietario le añadimos permiso de escritura, y a los usuarios que no sean el propietario o el grupo le quitamos la lectura
chmod u=rw fich	Al propietario le dejamos lectura, escritura, pero no ejecución
chmod g=u	Al grupo el ponemos los mismos permisos que tenga el
usuario *chmod u+s fich*	Activamos el bit set-uid
chmod u+r,a-r fich	¿Qué ocurre en este caso?

Cambio de usuario propietario

- Para modificar el propietario de ficheros o directorios se emplea el comando:

  chown [opciones] usuario[:grupo] archivos/directorios

- La opción más usada es “-R”, para aplicar el cambio de forma recursiva.
- El cambio de propietario se debe realizar con privilegios elevados.
- De forma opcional, permite cambiar también el grupo.

```
oper@oper-VirtualBox:~/Documentos/directorio$ ls -lrt
total 4
drwxrwxr-x 2 oper oper 4096 nov  4 10:24 dir_temp
--wx-wx--- 1 oper oper    0 nov  4 10:24 fichero
oper@oper-VirtualBox:~/Documentos/directorio$ sudo chown juan *
oper@oper-VirtualBox:~/Documentos/directorio$ ls -lrt
total 4
drwxrwxr-x 2 juan oper 4096 nov  4 10:24 dir_temp
--wx-wx--- 1 juan oper    0 nov  4 10:24 fichero
```

Cambio de grupo propietario

- Para modificar el grupo de ficheros o directorios se emplea el comando:

  chgrp [opciones] grupo archivos/directorios

- La opción más usada es “-R”, para aplicar el cambio de forma recursiva.

```
juan@oper-VirtualBox:/home/oper/Documentos/directorio$ ls -lrt
total 4
drwxrwxr-x 2 juan oper 4096 nov  4 10:24 dir_temp
--wx-wx--- 1 juan oper    0 nov  4 10:24 fichero
juan@oper-VirtualBox:/home/oper/Documentos/directorio$ chgrp grupojuan *
juan@oper-VirtualBox:/home/oper/Documentos/directorio$ ls -lrt
total 4
drwxrwxr-x 2 juan grupojuan 4096 nov  4 10:24 dir_temp
--wx-wx--- 1 juan grupojuan    0 nov  4 10:24 fichero
juan@oper-VirtualBox:/home/oper/Documentos/directorio$
```