Archivos y Directorios en Linux. Permisos

Gestión desde la terminal, mediante comandos

Árbol de directorios:

Cualquier dirección de archivo o carpeta en Linux empieza por el directorio raíz o /, seguido de todos los directorios y subdirectorios que lo contienen, separados cada uno de ellos por /

Gestión:

1. **cd**

cambiar de directorio

cd /home entrar en el directorio “home”.
cd .. retroceder un nivel.
cd / ir al directorio raíz.
cd ~user1: ir al directorio user1.
cd – ir (regresar) al directorio anterior.

2. **pwd**

mostrar el directorio actual de trabajo.

3. **ls**

ver los ficheros de un directorio.

ls -R: listado en forma de árbol, o recursiva.
ls -a: mostrar los ficheros ocultos.
ls *[0-9]*: mostrar los ficheros y carpetas que contienen números.

4. **tree**

mostrar los ficheros y carpetas en forma de árbol comenzando por la raíz.

5. **mkdir**

crear una carpeta o directorio

mkdir dir1: crear una carpeta o directorio con nombre ‘dir1’.
mkdir dir1 dir2: crea las dos carpetas o directorios simultáneamente
mkdir -p /tmp/dir1/dir2: crear un árbol de directorios.

6. **rm**

borrar fichero

rm file1: borrar el fichero llamado ‘file1’.

7. **rmdir**

borrar directorio

rmdir dir1: borrar la carpeta llamada ‘dir1’.
rm -rf dir1: eliminar una carpeta llamada ‘dir1’ con su contenido de forma recursiva. (Si lo borro recursivo estoy diciendo que es con su contenido).

8. **mv**

renombrar o mover un fichero o carpeta

mv dir1 new_dir: renombrar o mover un fichero o carpeta (directorio).

9. **cp**

copiar un fichero **/**directorio

cp file1 file2 copiar dos ficheros al unísono.

cp dir /* . copiar todos los ficheros de un directorio en el directorio de trabajo actual.

cp -a /tmp/dir1 . copiar un directorio dentro del directorio actual de trabajo.

cp -a dir1 copiar un directorio.

cp -a dir1 dir2 copiar dos directorio al unísono.

10. **ln**

crear un enlace simbólico al fichero o directorio

ln -s file1 lnk1: crear un enlace simbólico al fichero o directorio.

11.**touch**

crear fichero

Crear un fichero se puede hacer de varias formas distintas, una es la anterior, otras pueden ser mediante filtro de redirección, ...

Ejemplos vistos en clase:

a) echo "Esta es la primera línea del fichero" > fichero1.txt

b) cat > fichero2.txt

```
Esta es la primera línea de nuestro nuevo fichero
Esta es la segunda línea de nuestro nuevo fichero
y así sucesivamente iremos añadiendo contenido
hasta que queramos finalizar, en cuyo caso hay que pulsar CTRL+D
```

12.cat

mostrar contenido de un fichero

13.**head**

muestra filas superiores de un fichero. Si no se pone número son las 10 primeras.

14.**tail**

muestra filas inferiores de un fichero, indicando el número de líneas a mostrar

tail -n [número_filas] [nombre_archivo]

15.**less**

Permite **ver el contenido de un archivo de página a página**. Para salir del editor basta con pulsar la tecla q

16.**sort**

ordenar fichero

sort fichero1 ordena alfabéticamente por líneas

sort -r fichero1 ordena alfabéticamente en orden inverso por líneas

sort -n fichero1 ordena numéricamente por líneas

sort -c fichero1 comprueba si fichero1 ya está ordenado

Ejemplo: Supongamos que existe un archivo con una lista de coches llamada coches.txt.

Audi
Cadillac
BMW
Dodge

El comando sort puede utilizarse para saber si este archivo está ordenado y qué líneas están desordenadas.

sort -c coches.txt

sort: coches.txt:3: fuera de secuencia: BMW

Si no hay salida, se considera que el archivo ya está ordenado.

sort -u fichero1 ordena y elimina duplicados de fichero1

17.**grep**

Buscar contenido dentro de ficheros

grep Aug /var/log/messages: buscar palabra "Aug" en el fichero '/var/log/messages'.

grep [0-9] /var/log/messages: seleccionar todas las líneas del fichero '/var/log/messages' que contienen números.

Encontrar archivo(s)

find / -name file1: buscar fichero y directorio a partir de la raíz del sistema.

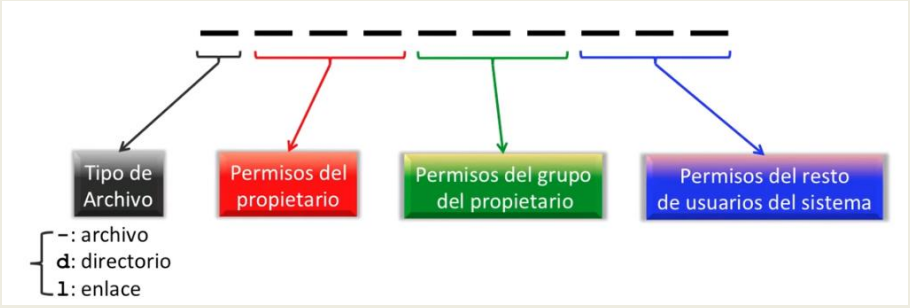
find / -user user1: buscar ficheros y directorios pertenecientes al usuario 'user1'.

find /home/user1 -name *.bin: buscar ficheros con extensión '. bin' dentro del directorio '/ home/user1'.

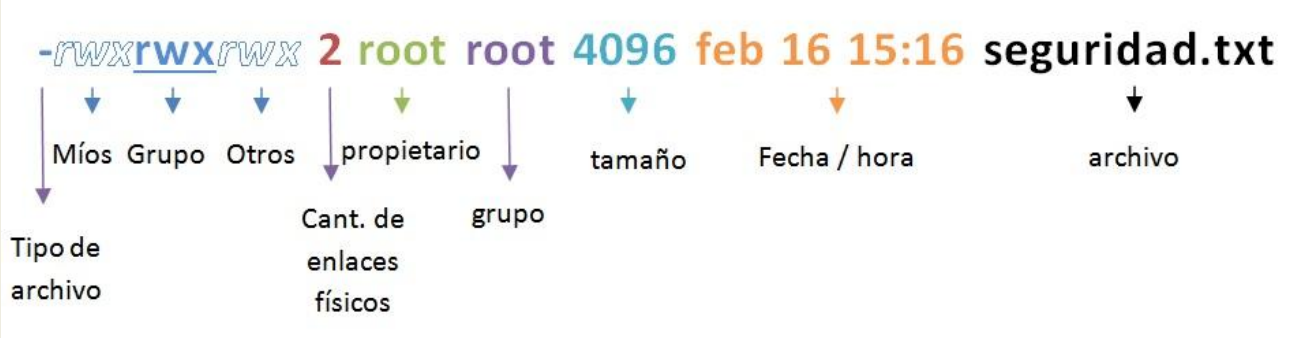
Permisos en ficheros (y directorios)

18. **ls -l**

muestra un listado donde se pueden observar los detalles de ficheros y carpetas, incluso los **permisos** (grupo de 9 caracteres)



Y un ejemplo de la línea completa de la salida del listado:



19.**chmod**

modificar los permisos. chmod ***"permisos" fichero***

Linux es un sistema operativo robusto y potente que permite cambiar los permisos de los archivos y directorios para que sólo algunos usuarios puedan **leerlos (R), escribirlos (W) o ejecutarlos (X)**. Es, además, un sistema multiusuario, por lo tanto, la tarea de añadir, modificar, eliminar y en general administrar usuarios se convierte en algo no solo rutinario, sino importante, además de ser un elemento de seguridad.

En el caso de archivos los permisos que pueden darse o quitarse son: (r) lectura, (w) escritura y (x) ejecución. En el caso de directorios los permisos son: (r) para listar los archivos, (w) para escribir, crear o borrar archivos y (x) para acceder a archivos del directorio.

¿Permisos a quién?

Los permisos se dividen en 3 partes, y son:

- **usuario (u)** (el dueño, también conocido como owner. Es el creador del archivo)
- **grupo (g)** (el grupo al que pertenece el archivo)
- **otros (o)**, (los que no son ninguno de los dos anteriores)

¿Qué permisos se pueden tener?

- Lectura (r o read): leer el archivo, o verlo
- Escritura (w o write): eliminar o editar el archivo
- Ejecución (x o eXecute): ejecutar el archivo en caso de que sea un ejecutable o script

En el caso de directorios los permisos son: (r) para listar los archivos, (w) para escribir, crear o borrar archivos y (x) para acceder a archivos del directorio.

Formatos que modifican los permisos
Como norma general se utilizan dos tipos de notaciones:

- simbólica: Hace uso del alfabeto (letras).
- octal: Establece permisos mediante números

Simbólica:

- La máscara simbólica está formada por tres códigos:

Clases de usuario	
Símbolo	Significado
u	propietario
g	grupo
o	otros
a	todos

Permisos	
Símbolo	Significado
r	lectura
w	escritura
x	ejecución

Operación	
Símbolo	Significado
+	añadir permisos
-	quitar permisos
=	asignar permisos

Octal:

Octal	Decimal	Permission	Representation
000	0 (0+0+0)	No Permission	---
001	1 (0+0+1)	Execute	--x
010	2 (0+2+0)	Write	-w-
011	3 (0+2+1)	Write + Execute	-wx
100	4 (4+0+0)	Read	r--
101	5 (4+0+1)	Read + Execute	r-x
110	6 (4+2+0)	Read + Write	rw-
111	7 (4+2+1)	Read + Write + Execute	rwx

Lectura tiene el valor de *4*
Escritura tiene el valor de *2*
Ejecución tiene el valor de *1*

Sólo se pueden cambiar los permisos de un archivo o un directorio si se es su **propietario** o el superusuario **root**.

Para hacerlo se utiliza el comando `chmod`, con la sintaxis:

```
chmod {a,u,g,o}{+,-}{r,w,x} fichero
```

Según la teoría vista anteriormente, ejemplos para asignar/denegar permisos a ficheros:

A) En modo **simbólico**:

- Quitar el permiso de escritura al grupo sobre el fichero tuArchivo.txt:
  chmod g-w tuArchivo.txt
- Darle permiso de lectura al grupo sobre el fichero tuArchivo.txt:
  chmod g+r tuArchivo.txt
- Quitar todos los permisos a todos sobre el fichero tuArchivo.txt:
  chmod ugo-rwx tuArchivo.txt

  También se puede con:
  chmod a-rwx tuArchivo.txt

Darle permiso de ejecución al grupo y de lectura a otros sobre el fichero tuArchivo.txt:

chmod g+x,o+r tuArchivo.txt

B) En modo **octal**:

- Dar todos los permisos al usuario, de escritura al grupo y ninguno a otros sobre el fichero tuArchivo.txt:

chmod 720 tuArchivo.txt

- Dar permiso de escritura al usuario, de ejecución al grupo y de lectura a otros sobre el fichero tuArchivo.txt:

chmod 214 tuArchivo.txt

- Dar los permisos de lectura y escritura al usuario, y solo lectura tanto al grupo como a otros sobre el fichero tuArchivo.txt:

chmod 644 tuArchivo.txt

- Dar todos los permisos a todos sobre el fichero tuArchivo.txt:
  chmod 777 tuArchivo.txt

Cambiar de propietario y/o grupo a un fichero

20.**chown**
cambiar al propietario de ficheros.
chown ***nuevopropietario fichero1***

21.**chgrp**
cambiar al grupo de ficheros.
chgrp ***nuevogrupo fichero1***

Se pueden hacer las dos operaciones "a la vez", de la forma:
chown ***nuevopropietario:nuevogrupo fichero1***
cambia usuario y grupo propietario de un fichero

Máscara de usuario.

La máscara de usuario (*umask*, abreviatura de ***u**ser **mask***) es una función que establece los permisos predeterminados para los nuevos archivos y directorios creados en el sistema. Puede establecerse en notación octal de tres o cuatro dígitos o bien en notación simbólica. Puede establecerse cualquier valor para *umask*, pero debe tomarse en consideración que ésta jamás permitirá crear nuevos archivos ejecutables.

Cuando se utiliza la notación octal de cuatro dígitos, el primer dígito siempre corresponde a los permisos especiales, pero el valor de éste siempre será 0.

En el caso de los directorios, el valor inicial suele ser de 0775.
Para los archivos el valor inicial suele ser de 0664.

22. ***umask*** Se ve (en notación octal) la máscara actual que se utiliza

El valor de la máscara de usuario, que se asigna ejecutando umask, corresponde a los **bits contrarios del permiso predeterminado que se quiera asignar**. Es decir, si por ejemplo se quiere asignar una máscara de usuario equivalente a 0775 (rwxrwxr-x), el valor de la máscara de usuario corresponderá a 0002 (el resultado de la operación 777 menos 775), que será lo mismo que definir u=rwx,g=rwx,o=rx.

Si por ejemplo queremos asignar una máscara de usuario equivalente a 0744 (rwxr--r--), el valor de la máscara de usuario corresponderá a 0033 (el resultado de la operación 777 menos 744), que será lo mismo que definir u=rwx,g=r,o=r.

Ejemplos de permisos regulares:

Valor octal	Valor umask	Permisos de Usuario	Permisos de Grupo	Permisos de Otros
0400	0377	r--	---	---
0440	0337	r--	r--	---
0444	0333	r--	r--	r--
0500	0277	r-x	---	---
0550	0227	r-x	r-x	---
0555	0222	r-x	r-x	r-x
0644	0133	rw-	r--	r--
0664	0113	rw-	rw-	r--
0666	0111	rw-	rw-	rw-
0700	0077	rwx	---	---
0711	0066	rwx	--x	--x
0707	0070	rwx	---	rwx
0744	0033	rwx	r--	r--
0750	0027	rwx	r-x	---
0755	0022	rwx	r-x	r-x
0775	0002	rwx	rwx	r-x
0777	0000	rwx	rwx	rwx

23.**stat**

stat fichero Nos muestra información variada de un fichero o directorio: tamaño, ubicación, y lo que más nos interesa en este momento: sus permisos.

```
juan@juan-VirtualBox:~$ stat ejer1.txt
  Fichero: ejer1.txt
  Tamaño: 11            Bloques: 8          Bloque E/S: 4096   fichero regular
Dispositivo: 805h/2053d Nodo-i: 429871      Enlaces: 1
Acceso: (0764/-rwxrw-r--)  Uid: ( 1002/ titular)   Gid: ( 1006/suplente)
Acceso: 2022-04-22 15:26:48.965232637 +0200
Modificación: 2022-04-22 15:26:36.573058763 +0200
      Cambio: 2022-04-26 10:30:09.215945728 +0200
    Creación: -
```