

---

## GESTIÓN DE ARCHIVOS Y DE PERMISOS EN LINUX

---

En esta práctica veremos algunas operaciones básicas sobre archivos que nos permitirán conocer propiedades básicas de los mismos. Estas operaciones básicas deberían ser útiles para diferenciar los sistemas de archivos propios de Linux y de Windows NT.

El sistema de ficheros propio de Linux es ext3 (<http://es.wikipedia.org/wiki/Ext3>) (o su sucesor, ext4, <http://es.wikipedia.org/wiki/Ext4> en las últimas versiones del kernel Linux). En los sistemas operativos Windows NT el sistema de ficheros es ntfs (<http://es.wikipedia.org/wiki/NTFS>).

A lo largo de la práctica veremos algunas de las propiedades anteriores.

Arranca la máquina virtual de Linux (Es independiente la versión)

1. En primer lugar, antes de empezar a gestionar los ficheros, vamos a comprobar los distintos dispositivos de almacenamiento que nuestro sistema Linux es capaz de reconocer. Abre una terminal. Dirígete a la carpeta `"/dev"`. Comprueba la utilidad del comando `"fdisk"` (`man fdisk`). En la carpeta `"/dev"` ejecuta el comando:

```
$sudo fdisk -l
```

Apunta la información que nos ofrece sobre el particionado del disco duro así como de su estructura física y lógica. El directorio de inicio ("`/`") ha sido montado a partir del dispositivo identificado como `"sda1"`; salvo que montemos nuevas unidades, contiene todos los ficheros y carpetas con que trabajaremos en la práctica.

2. Vamos a recuperar los tipos de ficheros que nos podemos encontrar en un sistema de ficheros de una máquina Linux. Dirígete al escritorio.

Ejecuta el siguiente mandato:

```
$touch prueba
```

¿Qué ha sucedido? Comprueba el cometido del comando "touch" por medio de "man touch" (o "info touch").

3. Vamos a comprobar ahora el tipo de fichero a que corresponde "prueba". Ejecuta "ls -l". De los 10 caracteres primeros que obtienes en su descripción (-rw-r--r--), el primero (-) corresponde con el tipo de fichero. Comprueba en la página [http://en.wikipedia.org/wiki/Unix\\_file\\_types](http://en.wikipedia.org/wiki/Unix_file_types) a qué tipo de fichero corresponde.

4. Una instrucción útil en Linux que nos permite saber qué tipo de fichero tenemos entre manos es "file". Ejecuta "file prueba". ¿Qué información te ha facilitado sobre el fichero?

5. Ejecuta nano sobre el fichero prueba (nano prueba). Cambia su contenido(incluye algo de texto). Apunta ahora el resultado de realizar "file prueba".

6. Otro comando Linux que también nos provee de información interesante sobre un archivo o carpeta es "stat". Ejecuta "man stat" para saber qué función realiza.

Ejecuta:

```
$stat index  
$ls -li index
```

Observa los distintos datos que nos facilitan ambos comandos sobre el fichero. ¿Cuáles son comunes? ¿Qué información guarda el sistema de archivos sobre cada fichero?

7. Otro elemento típico del sistema de ficheros de Linux son los directorios. En Linux los directorios están implementados de manera interna como archivos que contienen listas de archivos. Sitúate en tu carpeta de inicio (/home/alumno o directamente ~). Ejecuta:

```
$ls -l
```

¿Qué tipo de fichero es Escritorio (cuál es la primera letra de su descripción)? Comprueba su significado en [http://en.wikipedia.org/wiki/Unix\\_file\\_types](http://en.wikipedia.org/wiki/Unix_file_types).

Ejecuta también el comando:

```
$file Escritorio
```

¿Qué respuesta has obtenido?

Por último, ejecuta:

```
$stat Escritorio
```

Apunta el número de enlaces (fuertes) que existen a Escritorio.

8. Una aplicación que puede ser útil para encontrar ficheros en los sistemas Linux es "find". Puedes ejecutar "man find" para aprender algo más sobre la misma. El comando "find" debe tener privilegios para poder acceder a las distintas carpetas de nuestro sistema. Si quieres buscar un fichero en todo el sistema, deberías usarlo con "sudo". Si sólo quieres buscar un fichero en tu directorio personal, lo anterior no es necesario. Ejecuta:

```
$find /home/alumno -name index
```

¿Qué salida has obtenido?

```
$sudo find / -name gnome
```

¿A partir de qué directorio se ha producido la búsqueda?

9. Otros dos tipos de ficheros bastante comunes en los sistemas Linux son los dispositivos de bloques y de caracteres. Los mismos se utilizan para representar algunos dispositivos hardware tales como discos duros o tarjetas de sonido, o también las propias terminales del sistema.

Dirígete a la carpeta "/dev". Ejecuta el mandato:

```
$ls -l | less
```

Los sistemas de ficheros propios de Linux, ext3 y ext4, soportan una política de permisos basada en los permisos definidos por el estándar POSIX.1e, que también se conoce como "permisos tradicionales Linux". En la actualidad, un estándar más complejo, basado en la versión NFSv4 de "ACL" (listas de control de acceso), y similar al que se aplica en Windows ([http://wiki.linux-nfs.org/wiki/index.php/ACLs#NFSv4\\_and\\_Windows\\_ACLs](http://wiki.linux-nfs.org/wiki/index.php/ACLs#NFSv4_and_Windows_ACLs)) se está aplicando de forma experimental en ext4.

El sistema de permisos propio de Linux se basa en definir tres tipos de permisos (lectura, escritura y ejecución, ó r-w-x) para tres tipos de usuarios (el propietario, el grupo y el resto de usuarios de la máquina, ó u-g-o).

De ahí que siempre nos encontremos, al listar ("ls -l") el contenido de un directorio, con nueve letras que definen los permisos que se tienen sobre cada fichero y directorio:

( )rwxrwxrwx ( ) propietario grupo (datos adicionales) nombre  
123456789

En el caso anterior, el fichero presentado tendría todos los permisos posibles (rwxrwxrwx) para todos los usuarios de la máquina (veremos más adelante qué significa leer, escribir y ejecutar para cada tipo de fichero en Linux). Los caracteres 1,2,3 indican los permisos para el propietario, los caracteres 4, 5 y 6 indican los permisos para los restantes miembros del grupo, y los 7,8 y 9 indican los permisos para el resto de usuarios. Si alguno de los permisos no está asignado, en su lugar aparece un guión:

( )rwxr-xr-- ( ) propietario grupo (datos adicionales) nombre

Una breve descripción de lo que significa cada uno de los permisos sería la siguiente:

- r:      Sobre ficheros: Permiso para leer un fichero  
         Sobre directorios: Permiso para listar el contenido de un directorio
- w:      Sobre ficheros: Permiso para modificar un fichero  
         Sobre directorios: Permiso para crear y borrar ficheros y directorios contenidos en él
- x:      Sobre ficheros: Permiso para ejecutar un fichero o script  
         Sobre directorios: Permiso para acceder a un directorio

10. El comando que permite modificar los permisos de un fichero o directorio es "chmod". Puedes consultar su manual ("man chmod"). Los permisos se pueden modificar tanto por medio de caracteres como por el modo octal. Por el modo de caracteres, podemos especificar los permisos para el propietario (u), su grupo (g) o el resto de usuarios (o), y los permisos serán de la forma +rwx, -rwx ó =rwx. Por medio de + y - añadimos o eliminamos permisos, por medio de = asignamos unos nuevos. En el modo octal la notación es un poco distinta. El permiso de lectura se asocia con un 4, el de escritura con un 2 y el de ejecución con un 1. La asignación de permisos sale de la suma de los anteriores números. Por ejemplo, 6 es permiso de lectura y escritura, pero no de ejecución. 7 es igual a todos los permisos, y 0 es igual a ninguno. Por medio de ternas "657", asignamos los permisos del usuario, su grupo y el resto de usuarios respectivamente. Puedes encontrar en [http://es.wikipedia.org/wiki/Chmod#Asignaci.C3.B3n\\_de\\_permisos\\_en\\_el\\_comando\\_chmod](http://es.wikipedia.org/wiki/Chmod#Asignaci.C3.B3n_de_permisos_en_el_comando_chmod) una descripción más detallada.

11. Muévete al escritorio. Crea un fichero "mensaje" y escribe en el mismo un texto sencillo. Crea también tres ficheros de texto f1, f2 y f3 con el contenido que quieras. Comprueba con "ls -l" los permisos por defecto que han sido asignados a cada uno de los ficheros.

12. Deniega al "resto de usuarios" todos los permisos sobre f1, f2 y f3.

```
$chmod o-rwx f1 f2 f3
```

(El carácter "-" se utiliza para quitar los permisos, el carácter "+" para asignarlos).

Comprueba con "ls -l" los cambios ocurridos.

13. Deniega a todos los usuarios los permisos de lectura y ejecución de f1.

```
$chmod ugo-rx f1 (también es posible chmod a-rx f1)
```

Comprueba con "ls -l" que el cambio ha tenido lugar. Intenta acceder al fichero (por ejemplo con less). Apunta el resultado.

14. Concede permiso de lectura al propietario de f1:

```
$chmod u+r f1
```

Comprueba que el cambio ha tenido lugar. Trata de ejecutar "less f1" y anota el resultado.

15. Comprueba el permiso de escritura del propietario de f1. Asegúrate de que no lo tiene (si lo tiene elimínalo). Trata de editar el fichero por medio de nano y apunta el resultado.

16. Concede, con un solo mandato, permisos de lectura y ejecución y deniega el de escritura sobre f1 y f2 al grupo y al propietario, sin modificar el del resto de usuarios.

17. Concede, con un solo mandato, permisos de lectura y ejecución, y deniega el permiso de escritura sobre "mensaje" al grupo y al propietario, y deniega todos los permisos al resto de usuarios. Una posibilidad sería la siguiente, pero puedes explorar otras:

```
$chmod ug+rx,ug-w,o-rwx mensaje
```

18. Deniega todos los permisos a todos los usuarios sobre el fichero "mensaje". Intenta ejecutar el fichero. Apunta el resultado (observa que ni siquiera se intenta ejecutar el fichero, la gestión de permisos actúa antes).

19. Concede el permiso de lectura al propietario y al grupo de mensaje, el de escritura y ejecución al propietario y al resto de usuarios, y deniega el resto de permisos. Comprueba el resultado.

20. Sobre el fichero mensaje, concede lectura y escritura al propietario, ejecución al grupo y lectura y ejecución a otros, denegando el resto de permisos.

21. Crea en tu escritorio tres directorios (mkdir) llamados dir1, dir2 y dir3. Dentro de cada uno de ellos crea un fichero con el contenido que desees, con nombres fich11, fich22 y fich33. Copia en dir3 el fichero mensaje. Puedes comprobar por medio de "ls -lR" la estructura de directorios de que dispones ahora.

22. Deniega a todos los usuarios (puedes abreviarlo con "a") el permiso de lectura de dir1, el de escritura de dir2 y el de búsqueda (ejecución) de dir3. Comprueba que la operación se ha completado con éxito (ls -l).

23. Trata de listar el contenido del directorio dir1. Apunta el resultado.

24. Muévete a dir1. Muestra el contenido del fichero fich11 por medio de less. Apunta el resultado.

25. Sitúate en dir2 y muestra la lista de ficheros que contiene. Muestra el contenido del fichero fich22, intenta modificarlo y apunta qué ha ocurrido. Intenta crear un nuevo fichero en dir2 y borrar fich22. Relaciona lo sucedido con los permisos que tienes sobre dir2.

26. Sitúate en el Escritorio. Intenta mostrar la lista de ficheros que dir3 contiene. Anota el resultado. Intenta mostrar el contenido de fich33 y crear y borrar un fichero en dir3. Apunta el resultado.

27. Vamos ahora a crear un nuevo usuario en nuestra máquina. Utiliza el mandato:

```
$sudo adduser cuasi
```

Completa los datos necesarios.

28. Crea un nuevo directorio dir4. Dentro del mismo crea dos ficheros fich41 y fich42. Modifica los permisos de fich41 de tal modo que el usuario y su grupo tengan permisos de lectura y escritura, y el resto de usuarios no tenga ningún permiso.

29. Cambia la sesión al usuario cuasi creado:

```
$su cuasi
```

30. Comprueba y apunta el resultado al intentar leer (less) el fichero fich41.

31. Veamos ahora cómo podemos cambiar el propietario de un fichero. Ejecuta el mandato:

```
$chown cuasi fich41
```

Apunta el resultado. ¿Cómo puedes solucionarlo? Inténtalo. ¿Ha funcionado? Si no es así, vuelve otra vez a logarte como usuario "alumno" y completa la acción. ¿Te ha hecho falta ahora usar "sudo"?

32. Trata de acceder ahora al fichero "fich41". ¿Lo has conseguido? ¿Por qué?

33. Cambia el grupo de pertenencia del fichero fich41 a "cuasi" por medio del mandato:

```
$chgrp cuasi fich41
```

(Quizá debas usar "sudo" para completar el mandato anterior).

Trata de acceder de nuevo al fichero fich41 con less. ¿Lo ha conseguido?

34. Lógate como cuasi. Trata de acceder al fichero. ¿Lo has conseguido?

35. El mandato umask nos permite definir una máscara de usuario que se aplicará a todos los ficheros que se creen a partir de que la máscara sea definida. Puedes encontrar información sobre el mismo en <http://es.wikipedia.org/wiki/Umask>. Ejecuta la siguiente máscara:

```
$umask a-rwx
```

Desplázate al directorio de inicio del usuario cuasi (cd ~; cd \$HOME; cd /home/cuasi, ...). Crea un nuevo fichero fich43. Comprueba los permisos que se han asignado al mismo.

36. Crea una nueva máscara tal que solo el propietario tenga acceso (lectura) a los nuevos ficheros. Crea un fichero fich44 y compruébalo.

37. Vuelca el resultado de los anteriores mandatos en un fichero "mandatos\_practica\_17" y enlázalo junto con el informe desde tu sitio web.