Directorios y ficheros en linux

17/01/2020

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.

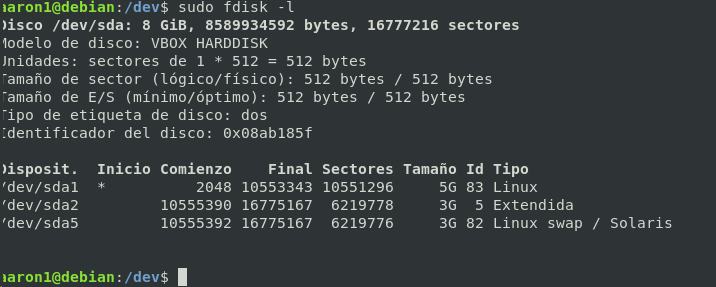
El sistema de ficheros propio de Linux es ext3 (<http://es.wikipedia.org/wiki/Ext3>) (o su sucesor, ext4, <http://es.wikipedia.org/wiki/Ext4>en las últimas versiones del kernel Linux). En los sistemas operativos Windows NT el sistema de ficheros es ntfs (<http://es.wikipedia.org/wiki/NTFS>). En la práctica que hicimos en el Tema anterior sobre Windows ya tuvimos oportunidad de comprobar algunas de las propiedades del sistema de ficheros ntfs. Algunas de las propiedades más relevantes de ext3 (y de ext4, <http://upload.wikimedia.org/wikipedia/commons/5/53/Tabla_ext3_ext4.png>) tienen que ver con su gestión por inodos (busca información sobre que es esto), con admitir enlaces débiles y fuertes, con el sistema de permisos que aceptan, o con el hecho de admitir (al igual que hace ntfs) “journaling”; “journaling” es una propiedad de ciertos sistemas de ficheros (y de bases de datos) por la cual los cambios hechos a un fichero no se aplican directamente sobre el mismo, sino que van a parar a un registro intermedio. En caso de que nuestro sistema falle, el sistema nos permitirá recuperar el registro de cambios realizados al mismo.

A lo largo de la práctica veremos algunas de las propiedades anteriores. Arranca la máquina virtual de Linux.

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 mandato “fdisk” (man fdisk). En la carpeta “/dev” ejecuta el mandato:

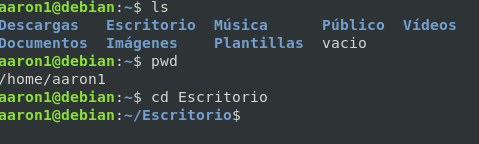
**$sudo fdisk –l**

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. El dispositivo “sda5” es utilizado como una partición “swap” o de intercambio (<http://es.wikipedia.org/wiki/Espacio_de_intercambio>).



Con el comando **fdisk –l** podemos ver el listado de discos montados que tenemos, para hacer esto primero tenemos que entrar en la carpeta dev y luego realizar **sudo** o a ver realizado anteriormente **sudo su**.

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

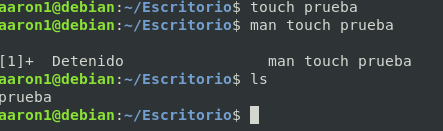


Para dirigirnos a Escritorio, tenemos que encontrarnos en el home y en el usuario, haciendo un **ls** vemos las carpetas y archivos que tenemos donde estamos, con un **pwd** podemos ver donde nos encontramos, este comando suele ser muy útil, con un **cd Escritorio** vamos al escritorio, Linux diferencia la mayúscula de las minúsculas, esto suele ser un dato muy importante a diferencia de Windows.

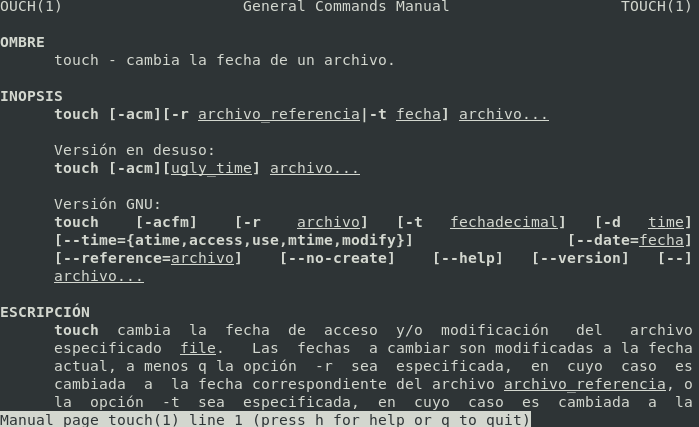
Ejecuta el siguiente mandato:

**$touch prueba**

¿Qué ha sucedido? Comprueba el cometido del mandato “touch” por medio de “man touch” (o “info touch”).

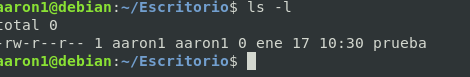


El comando **touch** lo usamos para crear un fichero, con el comando **touch prueba** hemos creado un archivo con el nombre de prueba.



Con el comando **man touch** obtenemos un manual del comando, esto suele ser muy útil

1. Vamos a comprobar ahora el tipo de fichero a que corresponde “prueba”. Ejecuta el mandato “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>a qué tipo de fichero corresponde.



Con el comando **ls –l** nos muestra el todos los archivos y carpetas con los permisos correspondientes de cada una.

En este caso tenemos un tipo de archivo regular, como puede ser un archivo de texto, esto lo sabemos con el primer carácter de los 10 que tenemos, los 9 siguientes son los permisos que tenemos.

1. Un mandato útil en Linux que nos permite saber qué tipo de fichero tenemos entre manos es el mandato file. Ejecuta el mandato “file prueba”.

¿Qué información te ha facilitado sobre el fichero?

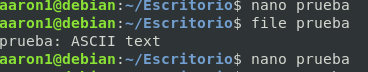


Con el comando **file prueba**, vemos el contenido que tenemos dentro de prueba, en este caso vemos que nos pone empty, esto quiere decir que el archivo está vacío.

1. Ejecuta nano sobre el fichero prueba (nano prueba). Cambia su contenido. Apunta ahora el resultado de realizar “file prueba”.

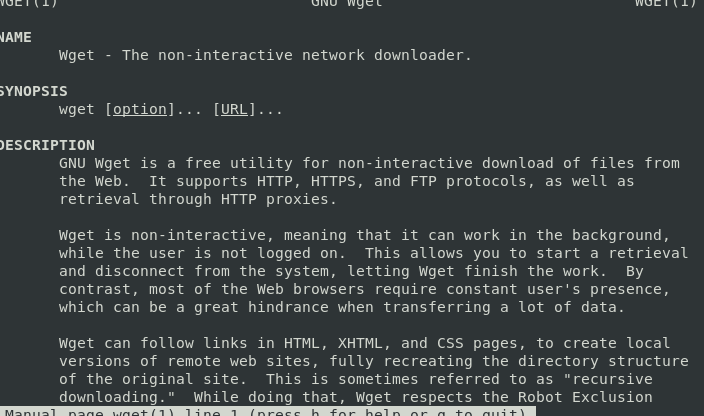


Con el comando **nano prueba** podemos entrar dentro del archivo y modificar su contenido, con **CONTRO O** guardamos y con **CONTROL X** salimos.



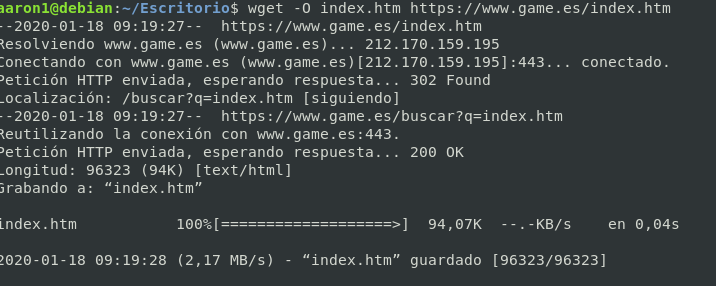
Ahora al realizar **file prueba**, vemos que pone ASCII text, esto quiere decir que tenemos un texto escrito en ASCII.

1. Vamos ahora a capturar una página web por medio del mandato “wget”. Puedes ejecutar “man wget” para saber algo más sobre este mandato. Ejecuta el siguiente mandato:



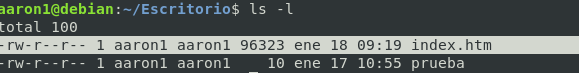
Con el comando **man wget**, vemos información sobre el comando.

**$wget –O index.htm** [**https://webdedescarga**](https://webdedescarga)**/index.htm**



La página WEB anterior, no funciona o ya no existe entonces he decidido coger otra cualquier en este caso la de GAME, con el comando **wget -O index.htm** [**https://www.game.es/index.htm**](https://www.game.es/index.htm)primero le estamos diciendo que con el nombre x se guarda en un archivo esa dirección.

Observa (por medio de “ls -l” o por medio del entorno de ventanas) que en tu escritorio ha aparecido un fichero de nombre “index.htm” que corresponde con la página de inicio de tu sitio web.



Con el comando **ls -l** vemos en los archivos y carpetas que tenemos dentro de Escritorio en este caso, vemos también los permisos de cada uno, aquí podemos ver que tenemos un archivo de tipo regular llamado index.htm

Vamos a ejecutar sobre el mismo el mandato:

**$file index.htm**

¿Qué respuesta obtienes?



Con el comando **file** vemos el tipo de archivo que es y el tipo de contenido que tiene, si realizásemos un **cat** podríamos ver con contenido de dentro de él, en este caso vemos que es tipo genérico y la configuración que tiene.

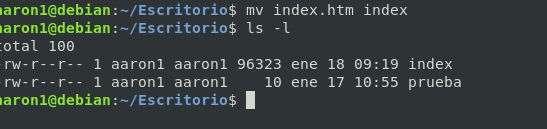


Podemos entrar en el contenido del archivo que hemos creado, en definitiva, el archivo es un html de la página web y dirección que le hemos dado.

1. El hecho de que el sistema de ficheros reconozca “index.htm” como un fichero de tipo “texto de HTML” no tiene nada que ver con la extensión del mismo. Ejecuta el siguiente mandato:

**$mv index.htm index**

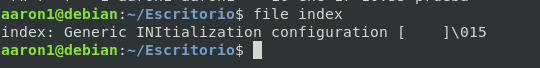
¿Qué ha sucedido? Compruébalo por medio de “ls –



Con el comando **mv** podemos ver el archivo.

Ejecuta ahora el mandato:

**$file index**

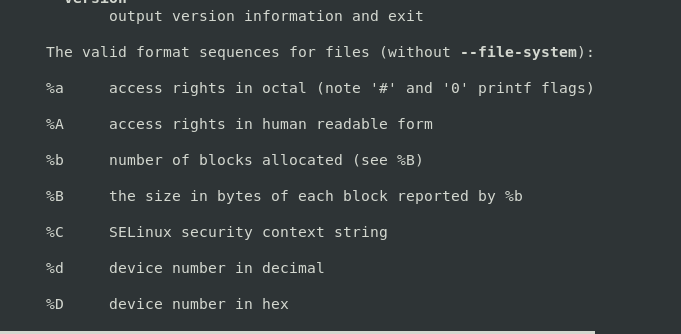


Vemos que el archivo antes se llamaba index.htm y ahora solamente index, el archivo sigue funcionando exactamente igual, ya que en Linux “no hay extensiones”.

¿Qué respuesta obtienes ahora? Recuerda los ejercicios que hicimos en Windows sobre las extensiones de los archivos y responde a la siguiente pregunta: ¿Qué hubiera sucedido en un sistema Windows si hubiéramos modificado la extensión de un archivo? ¿El sistema de ficheros seguiría reconociendo con qué tipo de aplicación debería abrirlo?

Obtengo la misma respuesta, pero con diferente nombre, Windows a diferencia de Linux si tiene extensiones y se la tenemos que dar por archivo que creemos, ya no aparecería la extensión en Windows y el archivo no sabría con qué tipo de aplicación la tendría que abrir, se lo tendríamos que decir nosotros.

1. Otro mandato Linux que también nos provee de información interesante sobre un archivo o carpeta es “stat”. Ejecuta “man stat” para saber qué función realiza.

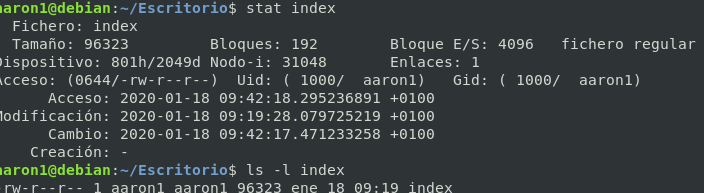


La función **stat** tiene dentro muchísimas más funciones, pero, en definitiva, nos da una información completa del fichero o carpeta, como el tamaño, bloques ocupados, tipo de fichero, los permisos actuales, etc. Para saber las funciones que realiza con el comando **man stat** podemos verlas.

Ejecuta los mandatos:

**$stat index**

**$ls –li index**



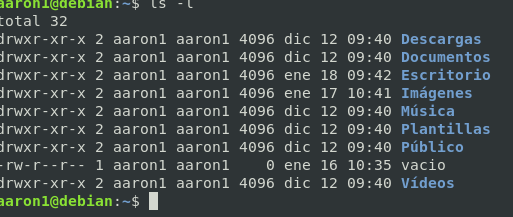
Observa los distintos datos que nos facilitan ambos mandatos sobre el fichero. ¿Cuáles son comunes? ¿Qué información guarda el sistema de archivos sobre cada fichero?

Vemos que con el comando **stat index** nos muestra toda la información de se archivó, tamaño, bloques, fichero regular, son los más comunes, guarda tanto a que usuario pertenece como al grupo del usuario al que pertenece, el tamaño del mismo y la fecha de creación. Esto se realiza con el comando **ls -l index** veremos la información básica

1. 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 el mandato:

**$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>.

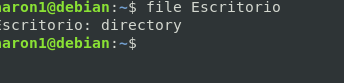


D es la primera letra de su descripción esto quiere decir que es un directorio.

Ejecuta también el mandato:

**$file Escritorio**

¿Qué respuesta has obtenido?

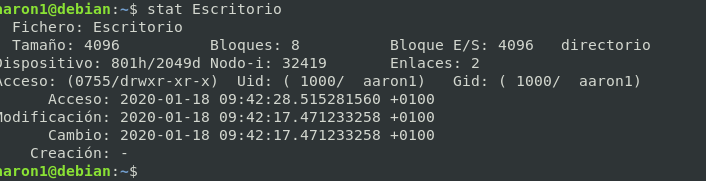


Podemos apreciar que nos dice que es un directorio, con el comando **file Escritorio**, tenemos que tener cuidado con las mayúsculas y minúsculas.

Por último, ejecuta el mandato:

**$stat Escritorio**

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



Tenemos dos Enlaces en el escritorio esto quiere decir que dentro de escritorio tenemos dos archivos



Una manera fácil de ver los archivos o directorios que tenemos dentro de un directorio seria, realizando el comando **dir**, el mismo que en Windows.

1. 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 mandato “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 los mandatos:

**$find /home/alumno –name index**

¿Qué salida has obtenido?



Aquí podemos ver la ruta de donde está el fichero llamado index.

**$sudo find / -name gnomine**

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



Se ha realizado a partir del directorio run.

1. Los dos tipos de ficheros anteriores (ficheros normales y directorios) no son los únicos que podemos encontrar en un sistema Linux. En la práctica anterior ya vimos cómo crear enlaces débiles y fuertes a un fichero. Desde el Escritorio crea un enlace débil al fichero “/usr/games/gnomine”.

**$ln –s /usr/games/gnomine enl\_debil\_gnomine**



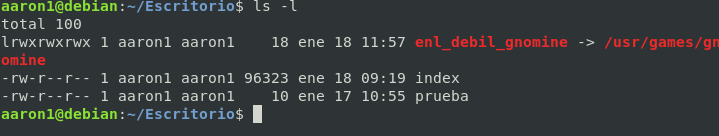
En el mismo Escritorio ejecuta ahora los mandatos:

**$fileenl\_debil\_gnomine** Apunta el resultado obtenido



**$ls –l**

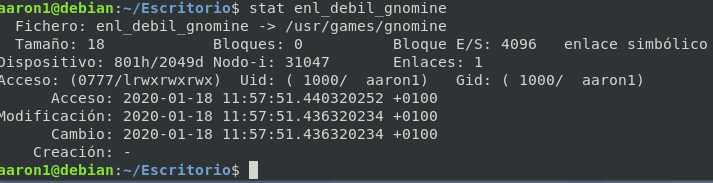
¿Qué tipo de fichero (cuál es el primer carácter) es el mismo (<http://en.wikipedia.org/wiki/Unix_file_types>)?



Es un fichero especial, esto lo sabemos por el primer carácter de los 10 que tenemos.

**$stat enl\_debil\_gnomine**

Localiza la información referente a los permisos del fichero.



Vemos que el fichero tiene permisos de lectura, escritura y ejecución tanto como para el usuario como para el grupo del usuario, como para todos los demás.

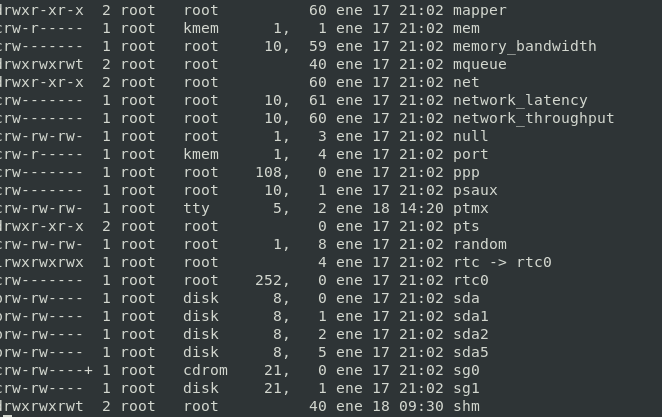
1. Otros dos tipos de ficheros bastante comunes en los sistemas Linux son los dispositivos de bloques y de caracteres. Los mismos se utilizan para representar algunos dispositivos hardware tales como discos duros o tarjetas de sonido, o también las propias terminales del sistema.

Dirígete a la carpeta “/dev”. Ejecuta el mandato:

**$ls –l | less**

Apunta algunos de los dispositivos cuyo nombre reconozcas (“tty…” corresponde con terminales, “sd…” corresponde con los dispositivos de almacenamiento…[http://en.wikipedia.org/wiki//dev/sda#Naming\_conventions](http://en.wikipedia.org/wiki/dev/sda#Naming_conventions)).

¿A qué tipo de fichero dentro de los propios de Linux corresponden (<http://en.wikipedia.org/wiki/Unix_file_types#Device_file>)?



Sda son dispositivos de almacenamientos montados o particiones, ptty son terminales.

Todos estos son archivos especiales, dvd, console también son algunos de los que conocemos.

Aún existen algunos otros tipos de ficheros propios (pipes, sockets…) en los sistemas Linux que aquí omitiremos. Pasamos ahora a ver algunas de las características del sistema de permisos propio de Linux.

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>) 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

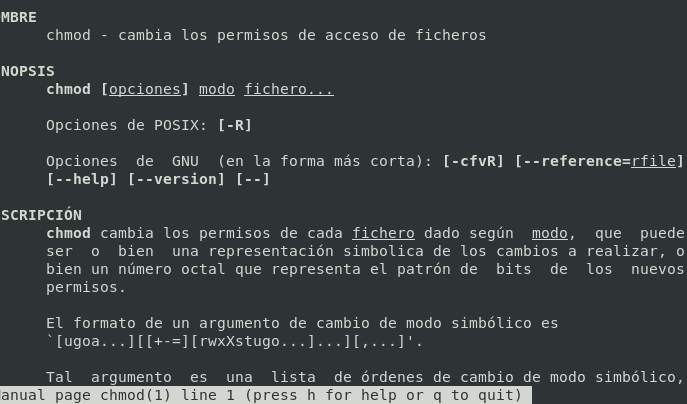
Sobre directorios: Permiso para listar el contenido de un directorio

w: Sobre ficheros: Permiso para modificar un fichero

Sobre directorios: Permiso para crear y borrar ficheros y directorios contenidos en él

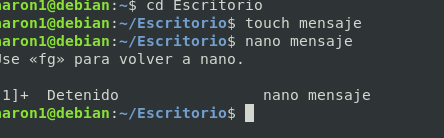
x: Sobre ficheros: Permiso para ejecutar un fichero o script Sobre directorios: Permiso para acceder a un directorio

1. Un mandato 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 lo 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> una descripción más detallada.



Con el comando **man chmod** vemos con detalle las funciones del comando **chmod**, el cual nos sirve para cambiar los permisos de los ficheros o directorios a usuarios, grupos o los demás, hay tres tipos de permisos que podemos dar, de lectura, escritura y de ejecución.

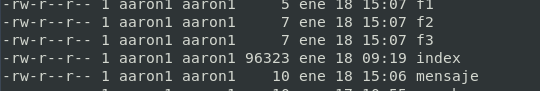
1. Muévete al escritorio. Crea un fichero “mensaje” y escribe en el mismo un texto sencillo. Crea también tres ficheros de texto f1, f2 y f3 con el contenido que quieras. Comprueba con “ls -l” los permisos por defecto que han sido asignados a cada uno de los ficheros.



Primero para poder ir al Escritorio escribimos **cd Escritorio**, una vez allí con el comando **touch** creamos un archivo y con nano podemos introducir o quitar líneas de texto.



Creamos a continuación los siguientes 3 ficheros de texto.



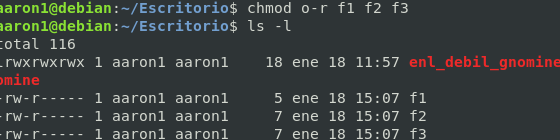
Vemos que los 4 ficheros que hemos creado tienen los mismos permisos, todos son archivos regulares, vemos que todos pueden escribir y leer, siendo su usuario, pero en grupo y a los demás solo pueden leer.

1. 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).

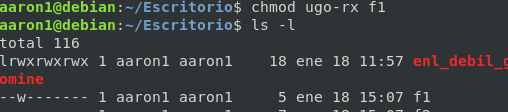
Comprueba con “ls -l” los cambios ocurridos



Vemos a continuación que con el comando **chmod o-r f1 f2 f3**, le hemos dicho que a todos esos ficheros le quiten el permiso de lectura a los demás usuarios , pudiendo todavía leer y escribir el creador de los ficheros.

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

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



Lo que hemos hecho con ese comando es muy sencillo le hemos de que al usuario, al grupo y a los demás le quiten el permiso de lectura y ejecución, pero no de escritura, también podíamos a verle dicho que solo al usuario ya que los demás ya no tenían permisos de ningún tipo.

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

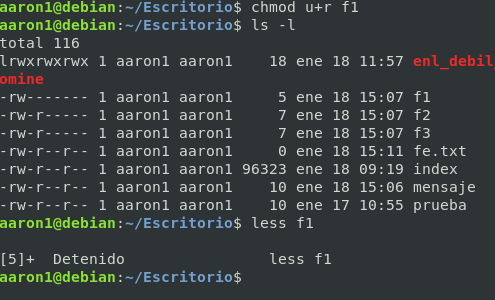


Vemos que al realizar un **less** nos deniega el acceso, porque no podemos leer el fichero.

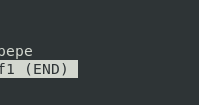
1. Concede permiso de lectura al propietario de f1:

**$chmod u+r f1**

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

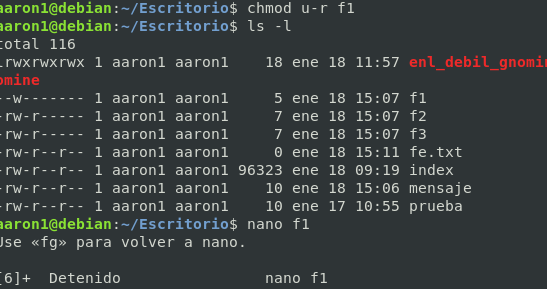


Vemos que tras a ver realizado un **chmod u+r f1** le hemos dicho que le demos permiso de lectura al usuario en el siguiente fichero.

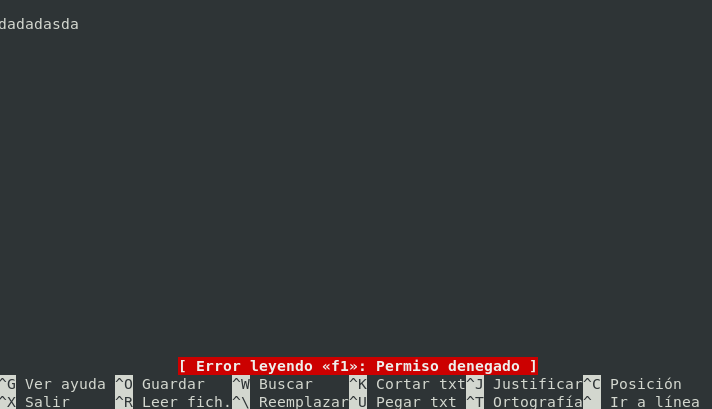


Después realizamos un **less f1** para comprobar que funciona y vemos como podemos acceder al texto escrito anteriormente.

1. 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.

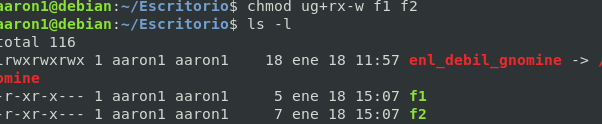


Realizamos un **chmod u-r f1** para quitar el permiso de lectura al usuario en el fichero, después realizamos un **ls -l**, para comprobar que se ha quitado correctamente, después realizamos un **nano f1**.



Podemos realizar un **nano** el fichero f1, porque como usuarios tenemos permiso de escribir, pero no de leer.

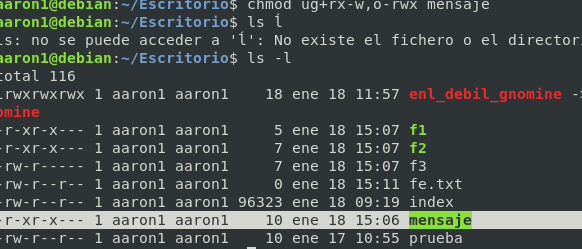
1. Concede, con un solo mandato, permisos de lectura y ejecución y deniega el de escritura sobre f1 y f2 al grupo y al propietario, sin modificar el del resto de usuarios.



Con **chmod ug+rx -w f1 f2** le decimos que a los ficheros f1 y f2 les demos permisos de lectura y ejecución tanto al usuario como al grupo, esto es con **ug+rx** y que le quitemos permisos de escritura al grupo y al propietario con **-w**.

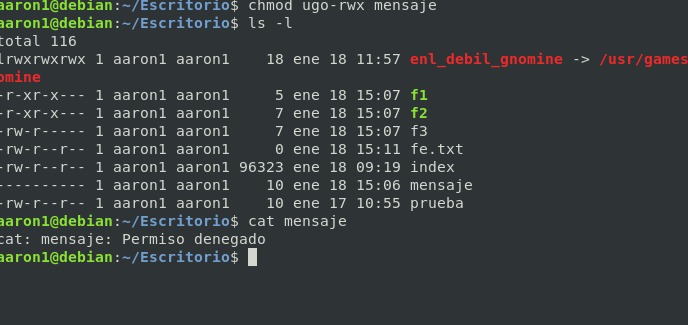
1. Concede, con un solo mandato, permisos de lectura y ejecución, y deniega el permiso de escritura sobre “mensaje” al grupo y al propietario, y deniega todos los permisos al resto de usuarios. Una posibilidad sería la siguiente, pero puedes explorar otras:

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



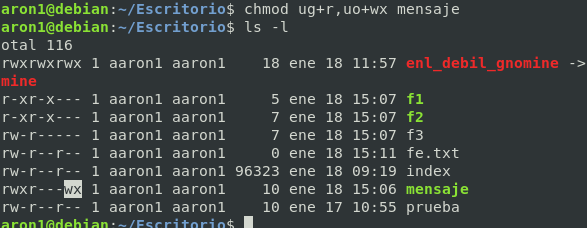
Con el comando **chmod ug+rx-w,o-rwx** mensaje le estamos diciendo que a el usuario y al grupo le de permisos de lectura y ejecución y que le quite permisos de escritura a los dos, y con **o-rwx** le estamos diciendo que a los demás le quitemos todos los permisos.

1. 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).



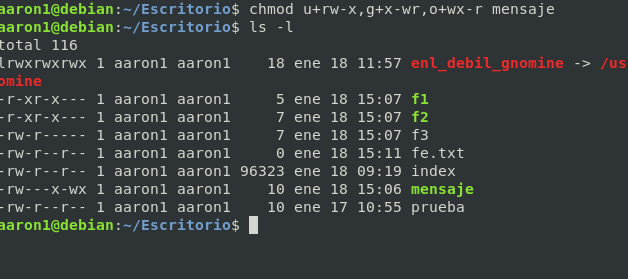
Para denegar los permisos a todos los usuarios realizamos un **chmod ugo -rwx mensaje** después realizamos un **ls -l** vemos que se han quitado todos los permisos correctamente y por último intentamos abrir el archivo y vemos que no podemos abrirlo, porque no tenemos permisos de ejecución.

1. 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.



Con el comando **chmod ug+r** concedemos permisos de lectura al usuario y al grupo, con **uo+wx** concedemos permisos de escritura y ejecución al usuario mas a los demás, todo esto en el fichero mensaje, a continuación, realizamos un **ls -l** para comprobar los cambios, no quitamos permisos, porque el fichero no tenía anteriormente ningún permiso concedido.

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

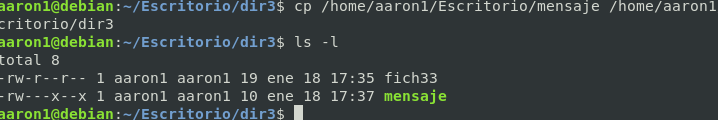


Con el comando **chmod** concedemos o quitamos permisos sobre un fichero o directorio a un usuario, grupo o otros.

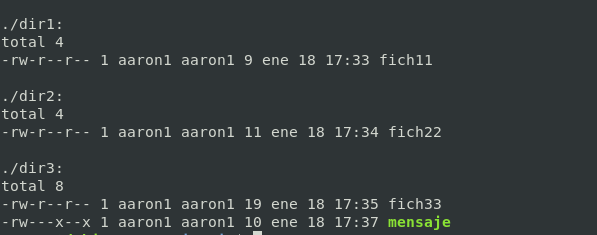
1. Crea en tu escritorio tres directorios (mkdir) llamados dir1, dir2 y dir3. Dentro de cada uno de ellos crea un fichero con el contenido que desees, con nombres fich11, fich22 y fich33. Copia en dir3 el fichero mensaje. Puedes comprobar por medio de “ls -lR” la estructura de directorios de que dispones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



Con el comando **mkdir** creamos directorios, en este caso creamos 3 directorios dentro de escritorio, para poder crear un archivo dentro de estos directorios, lo hacemos mediante el comando **touch** y a continuación añadimos el texto que queramos dentro del fichero creado, uno por directorio.

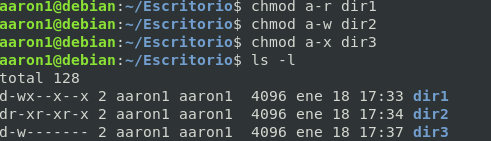


Con el comando **cp** copiamos un archivo, en este caso queremos copiar el fichero mensaje en el dir3, para hacer esto escribimos primero **cp** la ruta de donde esta el fichero que queremos copiar y luego donde lo queremos copiar y a continuación realizamos un **ls -l** para comprobar que se ha copiado correctamente.



Para poder ver tanto la estructura de directorios y el contenido de cada uno realizamos un **ls -lR** vemos que tenemos 3 directorios dentro de escritorio dir1 dir2 y dir3 en cada uno hay un fichero, pero en el dir3 tenemos también el fichero mensaje que hemos copiado anteriormente con el comando **cp**, podemos ver que a la hora de copiar un fichero también se copian los permisos que tiene en ese momento.

1. 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).



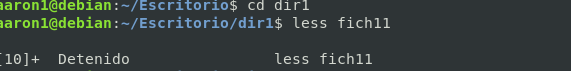
Para denegar o dar permisos a fichero utilizamos el comando **chmod**, en este caso le decimos con a que, a todos los usuarios, ya sean de grupos u otros, les deneguemos o demos permisos, luego realizamos un **ls -l** para comprobar que todos los permisos se han quedado correctamente.

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

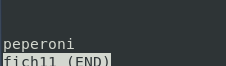


No podemos ver el contenido que tenemos en el dir1 porque no tenemos permisos, solo como usuarios principales podemos escribir o ejecutar, para poder ver el contenido de un fichero usamos el comando **ls**.

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

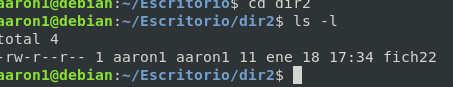


Para poder movernos dentro del dir1 usamos el comando **cd dir1**, después utilizamos un **less fich11** para ver que contenido tenemos dentro del fich11.

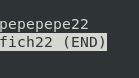


Vemos que podemos leer el fich11, perfectamente.

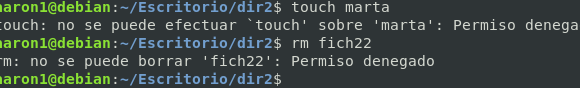
1. Sitúate en dir2 y muestra la lista de ficheros que contiene. Muestra el contenido del fichero fich22, intenta modificarlo y apunta qué ha ocurrido. Intenta crear un nuevo fichero en dir2 y borrar fich22. Relaciona lo sucedido con los permisos que tienes sobre dir2.



Para situarnos en dir2, realizamos un **cd dir2**, a continuación, realizamos un **ls – l** para ver todos los ficheros que tenemos en este directorio.



Vemos que al realizar un **less** no tenemos ningún problema ya que tenemos permisos de lectura y realizando un **nano fich22** podemos modificar perfectamente el archivo.

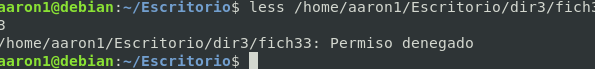


Vemos que no tenemos permisos de creación ni permisos para eliminar ningún archivo, en dir2 que es el directorio donde nos encontramos no tiene permisos de escritura, pero si de lectura por eso podemos realizar un **ls -l** a diferencia del dir1, pero no podemos escribir entonces no podemos crear ni eliminar nada solo ejecutar y leer.

1. 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.



No podemos ver los ficheros que tiene el directorio dir3, porque no tenemos permisos de lectura en dir3 solo tenemos permisos de escritura en el usuario principal y en los grupos y otros no tenemos ningún tipo de permiso.



Tampoco podemos ver el contenido de fich33 tampoco tenemos permisos de lectura.

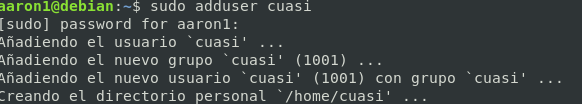


Tampoco podemos crear ningún tipo de archivo ni borrar ninguno.

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

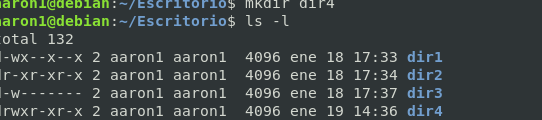
**$sudo adduser cuasi**

Completa los datos necesarios.

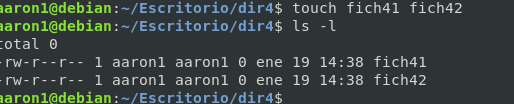


Para poder crear un usuario utilizamos el comando **sudo adduser cuasi**, después tendremos que crear una contraseña para ese usuario su nombre y datos básicos.

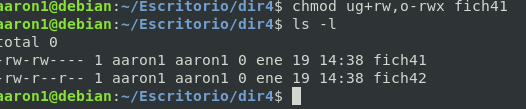
1. Crea un nuevo directorio dir4. Dentro del mismo crea dos ficheros fich41 y fich42. Modifica los permisos de fich41 de tal modo que el usuario y su grupo tengan permisos de lectura y escritura, y el resto de usuarios no tenga ningún permiso.



Creamos un nuevo directorio dentro de Escritorio, le llamamos dir4, creamos el directorio con el comando **mkdir** después realizamos un **ls -l** para comprobar que se ha creado correctamente y que tipos de permisos tiene.



Después creamos los dos ficheros a la vez dentro de dir4, para ello usamos el comando **touch** y después realizamos un **ls -l** para comprobar que se han creado correctamente.



Para conceder o quitar permisos lo hacemos con el comando **chmod** escribimos **ug+rw** para decirle que nos de permisos de escritura y lectura tanto al usuario como al grupo y después realizamos un **o-rwx** para quitar todos los permisos a los demás usuarios.

1. Cambia la sesión al usuario cuasi creado:

**$su cuasi**



Entramos en el usuario cuasi creado anteriormente, esto lo hacemos con el comando **su cuasi**.

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



No podemos leer el contenido de fich41, porque otros usuarios no tienen permisos para leer solo, el grupo del usuario y el usuario.

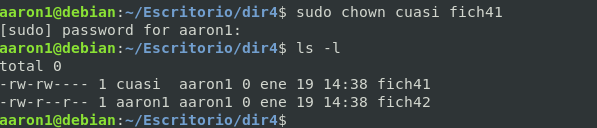
1. Veamos ahora cómo podemos cambiar el propietario de un fichero. Ejecuta el mandato:

**$chown cuasi fich41**



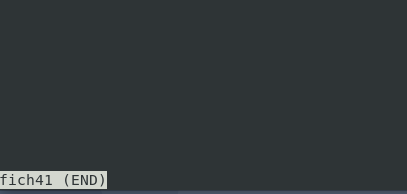
No podemos realizar esa opción como no somos propietario del archivo.

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”?

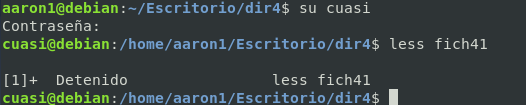


Para poder solucionarlo, yo he vuelto a entrar a el usuario aaron1 y le he dicho con **sudo chown cuasi fich41** que quiero que en el archivo fich41 se cambien el propietario en este caso que sea cuasi, después realizamos un **ls -l** para comprobar que se ha cambiando el propietario del archivo correctamente, me ha hecho falta realizar sudo porque es un tipo de permiso administrador.

1. Trata de acceder ahora al fichero “fich41”. ¿Lo has conseguido? ¿Por qué?



Vemos que hemos podido acceder ahorra correctamente al fichero con el comando **less**, a continuación, veremos cómo se hace.

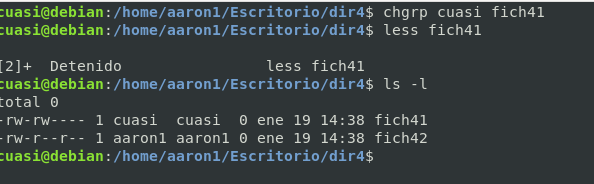


Primero tenemos que volver al usuario cuasi, con el comando **su cuasi** vamos al usuario, después de poner la contraseña, realizamos un **less fich41**, vemos que ahora funciona, esto pasa, porque hemos cambiado el usuario propietario del fichero.

1. 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).



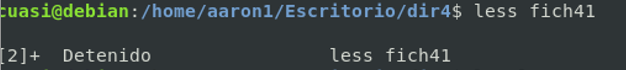
Para cambiar un fichero de grupo necesitamos usar el comando **chgrp** en este caso **chgrp cuasi fich41** le decimos que queremos que nos cambien el fichero a ese grupo. No tenemos ningún tipo de problema al cambiarlo, porque ya somos propietario del fichero. Después realizamos un **ls -l** para comprobar como se ha cambiado el grupo del fich41.

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

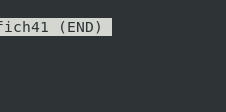


Vemos que no podemos acceder al archivo perfectamente, realizando un **less fich41**.

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



Si podemos entrar perfectamente.

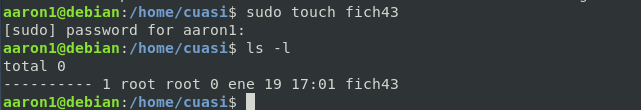


1. El mandato umask nos permite definir una máscara de usuario que se aplicará a todos los ficheros que se creen a partir de que la máscara sea definida. Puedes encontrar información sobre el mismo en <http://es.wikipedia.org/wiki/Umask>. Ejecuta la siguiente máscara:

**$umask a-rwx**

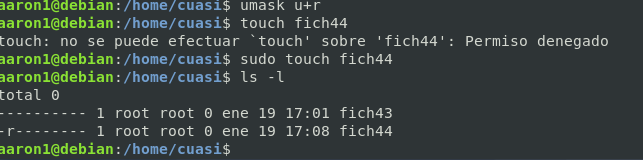
Desplázate al directorio de inicio del usuario cuasi (cd ~; cd $HOME; cd

/home/cuasi, …). Crea un nuevo fichero fich43. Comprueba los permisos que se han asignado al mismo.



Una vez que nos encontramos en el home de cuasi, realizando un **cd** mas la ruta podremos situarnos en cuasi, realizamos un **sudo touch fich43** ponemos la contraseña y realizamos un **ls -l**, vemos que no tenemos permisos de ningún tipo, porque hemos realizado un **umask a-rwx** lo que hacemos con este comando es que a partir de que lo hayamos ejecutado todos los demás ficheros van a ser creado con esa norma de permisos en este caso, ningún usuario tiene permiso a anda.

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



Con el comando **umask u+r** le decimos que a partir de ese comando todos los demás archivos van a tener permisos de lectura solamente por el propietario, no tenemos que quitar los permisos a todos los demás, porque no los tenía anteriormente, después realizamos un sudo **touch fich44** para crear el fichero y por último un **ls -l** para comprobar que los permisos se han puesto correctamente.

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

Con respecto a Windows, es importante que recuerdes algunas de las propiedades del sistema de archivos NTFS que vimos en las primeras prácticas del curso. En la parte referente a permisos, desde el intérprete cmd se pueden modificar los permisos de ficheros y directorios por medio de la herramienta conocida como cacls, aunque la misma no dispone de la funcionalidad de la herramienta propia de la GUI. Puedes comprobar algunas de sus características en <http://en.wikipedia.org/wiki/Cacls>.