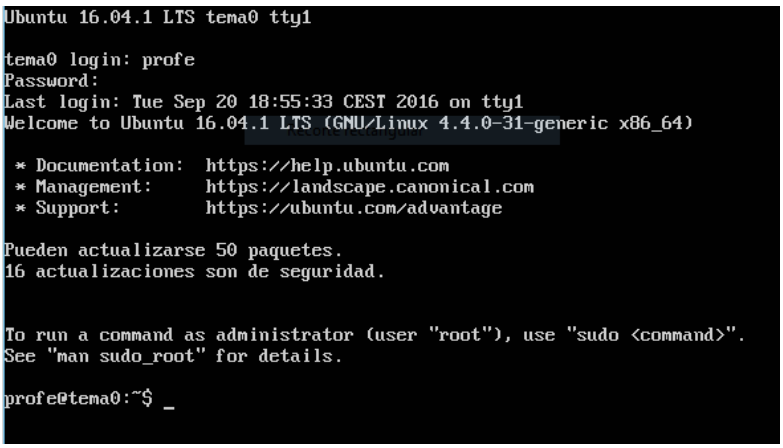
1. ENTRADA Y SALIDA AL SISTEMA (LOGIN)

Para poder usar Linux, lo primero que hay que hacer es identificarse con un **nombre de usuario** y una **contraseña.**

El nombre de usuario no puede contener caracteres especiales como signos de puntuación (, ; :), la barra invertida (/), etc. La clave debe ser suficientemente larga y difícil de adivinar. No es buena idea utilizar como clave el nombre, apellidos, el número de teléfono, el número de la tarjeta de crédito o un nombre de mascota. Si la clave que utiliza un usuario es corta o fácil de adivinar corre el riesgo de que alguien entre en su sistema y borre o modifique información importante.

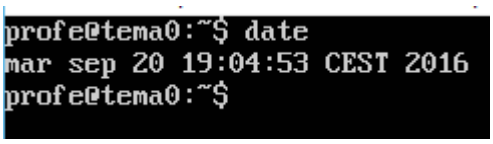


La contraseña no aparece por pantalla mientras se teclea. Hay que tener cuidado con las mayúsculas y las minúsculas, si el sistema dice que la clave no es correcta puede que esté activada la tecla "BlqMayús". Una vez introducidos el nombre de usuario y la clave, si el proceso de login se lleva a cabo correctamente, el sistema muestra el prompt con el formato:

nombre_de_usuario@nombre_de_la_máquina:~$

En este caso, el nombre de usuario es "profe", el nombre de la máquina es "tema0" y aparece un carácter "$" que indica que el usuario conectado es un usuario "normal". Cuando un usuario tiene privilegios de root (super-usuario) aparece el carácter "#" como se verá más adelante.

¡Linux ya está listo para ejecutar comandos! El lector puede probar con el comando "date" para ver la fecha actual.



Para salir del sistema podemos usar los comandos

poweroff

exit

Para activar la cuenta root tienes que usar el comando **sudo passwd root**, meter la contraseña de tu usuario (en este caso el mío es profe), y luego asignar una contraseña para root.

```
profe@tema0:/etc/network/interfaces.d$ sudo passwd root
[sudo] password for profe:
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
profe@tema0:/etc/network/interfaces.d$ _
```

Para acceder como root puedes iniciar sesión como usuario root o acceder desde el usuario en el que te encuentras con el comando **su root** (además con el comando **sudo** este comando también puedes realizar cualquier acción de superusuario desde tu propio usuario. Por ejemplo: sudo apt-get install tree y luego poniendo la contraseña de root).

```
profe@tema0:/etc/network/interfaces.d$ su root
Contraseña:
root@tema0:/etc/network/interfaces.d#
```

2. ESTRUCTURA DE DIRECTORIOS.

Trabajando en el entorno gráfico se habla de carpetas y trabajando con comandos en un terminal, se habla de **directorios,** pero conceptualmente son exactamente lo mismo.

A continuación se muestra una tabla con **los directorios más importantes** de un sistema Linux:

/bin	Contiene programas ejecutables básicos para el sistema.
/boot	Contiene los ficheros necesarios para el arranque del sistema.
/dev	Contiene los ficheros correspondientes a los dispositivos: sonido, impresora, disco duro, lector de cd/dvd, video, etc.
/etc	Contiene ficheros y directorios de configuración.
/home	Contiene los directorios de trabajo de los usuarios. Cada usuario tiene su propio directorio en el sistema dentro de /home/.
/lib	Contiene las librerías compartidas y los módulos del kernel
/media	Dentro de este directorio se montan los dispositivos como el CD-ROM, memorias USB, discos duros portátiles, etc
/opt	Directorio reservado para instalar aplicaciones.
/sbin	Contiene los ficheros binarios ejecutables del sistema operativo.
/srv	Contiene datos de los servicios proporcionado por el sistema.
/tmp	Directorio de archivos temporales.
/usr	Aquí se encuentran la mayoría de los archivos del sistema, aplicaciones, librerías, manuales, juegos... Es un espacio compartido por todos los usuarios.
/var	Contiene archivos administrativos y datos que cambian con frecuencia: registro de errores, bases de datos, colas de impresión, etc.
/root	Directorio de trabajo del administrador del sistema (usuario root).
/proc	Aquí se almacenan datos del kernel e información sobre procesos.

3. VIZUALIZACIÓN, CREACIÓN Y CAMBIO DE DIRECTORIO (pwd, ls, cd, mkdir)

~ pwd

El comando pwd muestra cuál es el directorio de trabajo actual, en otras palabras, le dice al usuario dónde se encuentra dentro de la estructura de directorios del sistema. Es muy útil cuando estamos **p**er**d**idos.

```
profe@tema0:~$ pwd
/home/profe
profe@tema0:~$ _
```

~ ls

El comando ls muestra el contenido del directorio actual. Por defecto, los archivos ocultos no se muestran. Éste es seguramente el comando que más se utiliza.

```
profe@tema0:/etc/network$ pwd
/etc/network
profe@tema0:/etc/network$ ls
if-down.d  if-post-down.d  if-pre-up.d  if-up.d  interfaces  interfaces.d
profe@tema0:/etc/network$ _
```

Se pueden añadir opciones a ls, por ejemplo:

ls –a muestra todos los archivos, incluyendo los ocultos (cuyo nombre comienza por un punto),
ls –l muestra un listado detallado, con la última fecha de modificación de cada archivo, el tamaño, etc.

ls –h muestra el tamaño de los ficheros en bytes, Kb, Mb, etc.

Todas las opciones disponibles, tanto para ls como para el resto de comandos se pueden consultar mediante las páginas del manual, con el comando man seguido del comando del que se quiere obtener información:

```
profe@tema0:/etc/network$ man ls
```

Esto dará información detallada sobre el comando ls . Para salir del manual basta pulsar la letra "q".

~ cd

El comando cd (change dir) permite cambiar de directorio. Si se utiliza tal cual, sin ningún tipo de argumento, cambia al directorio de trabajo personal. Si se utiliza seguido de una **ruta,** cambia al directorio que se indica.

```
profe@tema0:~$ pwd
/home/profe
profe@tema0:~$ cd /etc/network/
profe@tema0:/etc/network$
```

Las rutas pueden ser **absolutas** o **relativas.** Una ruta es absoluta cuando comienza por el carácter "/" y relativa cuando comienza por cualquier otro carácter.

En el ejemplo anterior se ha usado una ruta absoluta, esto es, /etc.

Veamos cómo cambiar a otros directorios utilizando otras rutas absolutas:

```
profe@tema0:/etc/network$ cd /dev/net/
profe@tema0:/dev/net$ _
```

En este ejemplo, el usuario profe estaba en el directorio /etc/network y ha ido al directorio /dev/net que NO ESTÁ DENTRO DE /etc/network. Por tanto hemos usado una **ruta absoluta** para acceder a él (poniendo delante de todo el carácter "/").

Veamos como se usa una ruta relativa:

```
profe@tema0:/etc/network$ cd interfaces.d/
profe@tema0:/etc/network/interfaces.d$ _
```

En este caso, el usuario estaba en /etc/network y ha accedido al directorio interfaces.d que está dentro del directorio network. Por tanto, ha usado una **ruta relativa** dentro de la ruta donde ya estaba (no usa después de cd el carácter "/").

Las rutas, tanto las absolutas como las relativas se pueden utilizar en la mayoría de comandos. No son algo específico que se utilice sólo con cd.

Dos puntos (..) hacen referencia al directorio que hay justo a un nivel superior.

cd .. sube un nivel en la estructura de directorios.

~ **mkdir**

Se pueden crear directorios con el comando mkdir. Por ejemplo, para crear una estructura de carpetas donde el usuario guardará información sobre su música, películas y libros, se hará la siguiente secuencia de comandos (quedando la estructura que hay a la derecha):

```
profe@tema0:~$ mkdir musica
profe@tema0:~$ mkdir peliculas
profe@tema0:~$ mkdir libros
profe@tema0:~$ cd musica
profe@tema0:~/musica$ mkdir decada60
profe@tema0:~/musica$ mkdir decada70
profe@tema0:~/musica$ mkdir decada80
profe@tema0:~/musica$ mkdir contemporanea
profe@tema0:~/musica$ cd ..
profe@tema0:~$ cd peliculas
profe@tema0:~/peliculas$ mkdir clasicas
profe@tema0:~/peliculas$ mkdir animacion
profe@tema0:~/peliculas$ mkdir accion
profe@tema0:~/peliculas$ cd ..
profe@tema0:~$ cd libros/
profe@tema0:~/libros$ mkdir poesia
profe@tema0:~/libros$ mkdir ensayo
profe@tema0:~/libros$ mkdir novela
profe@tema0:~/libros$ _
```

```
profe@tema0:~$ tree
.
├── libros
│   ├── ensayo
│   ├── novela
│   └── poesia
├── musica
│   ├── contemporanea
│   ├── decada60
│   ├── decada70
│   └── decada80
└── peliculas
    ├── accion
    ├── animacion
    └── clasicas

13 directories, 0 files
```

4. VISUALIZACIÓN DE FICHEROS (cat, more, less, head, tail)

Los comandos **cat, more y less** sirven para mostrar el contenido de ficheros de texto. La diferencia radica en cómo se muestra el contenido. A todos estos comandos hay que pasarles como argumento el fichero que se quiere mostrar. Se puede indicar una ruta, en caso de que el fichero que se quiere mostrar no esté en el directorio actual.

~ cat

El comando **cat** muestra por pantalla el contenido de un fichero y, cuando termina, el usuario está otra vez de vuelta en la línea de comandos. Por ejemplo:

```
profe@tema0:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet dhcp
profe@tema0:~$
```

Hemos mostrado el contenido del archivo interfaces que está dentro del directorio /etc/network/ (interfaces es un archivo, si fuera un directorio, no se podría mostrar con cat).

~ more

El comando **more** hace lo mismo que cat, a diferencia de que muestra el fichero pantalla a pantalla, es decir, llena de texto la pantalla y se espera a que el usuario pulse la tecla <espacio> para pasar a la siguiente.

~ less

El comando **less** es el más versátil de los tres, ya que permite moverse hacia delante y hacia atrás dentro del fichero, utilizando los cursores o las teclas de "AvPág" y "RePág".

En cualquier momento se puede interrumpir la visualización y volver al símbolo del sistema pulsando la letra "q".

~ head y tail

Los comandos head y tail permiten mostrar de forma parcial el contenido de un fichero. Como su nombre indica, head muestra las primeras líneas del fichero (la cabecera) y tail muestra las últimas líneas (la cola).

Ejemplo de head:

```
profe@tema0:/var/log$ head auth.log
Sep 20 18:42:11 tema0 systemd-logind[1996]: New seat seat0.
Sep 20 18:42:11 tema0 systemd-logind[1996]: Watching system buttons on /dev/input/event0 (Power Butt
on)
Sep 20 18:42:11 tema0 systemd-logind[1996]: Watching system buttons on /dev/input/event1 (Sleep Butt
on)
Sep 20 18:42:11 tema0 systemd-logind[1996]: Watching system buttons on /dev/input/event3 (Video Bus)
Sep 20 18:44:00 tema0 login[2240]: pam_unix(login:session): session opened for user profe by LOGIN(u
id=0)
Sep 20 18:44:00 tema0 systemd: pam_unix(systemd-user:session): session opened for user profe by (uid
=0)
Sep 20 18:44:00 tema0 systemd-logind[1996]: New session 1 of user profe.
Sep 20 18:55:28 tema0 login[2240]: pam_unix(login:session): session closed for user profe
Sep 20 18:55:28 tema0 systemd-logind[1996]: Removed session 1.
Sep 20 18:55:33 tema0 login[2365]: pam_unix(login:session): session opened for user profe by LOGIN(u
id=0)
profe@tema0:/var/log$
```

Ejemplo de tail:

```
profe@tema0:/var/log$ tail auth.log
Sep 20 19:46:06 tema0 su[2659]: + /dev/tty1 profe:profe
Sep 20 19:46:06 tema0 su[2659]: pam_unix(su:session): session opened for user profe by profe(uid=100
0)
Sep 20 19:46:06 tema0 su[2659]: pam_systemd(su:session): Cannot create session: Already running in a
 session
Sep 20 19:46:14 tema0 sudo:    profe : TTY=tty1 ; PWD=/etc/network/interfaces.d ; USER=root ; COMMAN
D=/usr/bin/apt-get install tree
Sep 20 19:46:14 tema0 sudo: pam_unix(sudo:session): session opened for user root by profe(uid=0)
Sep 20 19:46:17 tema0 sudo: pam_unix(sudo:session): session closed for user root
Sep 20 20:05:01 tema0 CRON[2944]: pam_unix(cron:session): session opened for user root by (uid=0)
Sep 20 20:05:01 tema0 CRON[2944]: pam_unix(cron:session): session closed for user root
Sep 20 20:17:01 tema0 CRON[2966]: pam_unix(cron:session): session opened for user root by (uid=0)
Sep 20 20:17:01 tema0 CRON[2966]: pam_unix(cron:session): session closed for user root
profe@tema0:/var/log$ _
```

Por defecto, tanto head como tail muestran 10 líneas, pero eso se puede cambiar con la opción –n + el número de líneas que queramos mostrar.

```
profe@tema0:/var/log$ head -n4 auth.log
Sep 20 18:42:11 tema0 systemd-logind[1996]: New seat seat0.
Sep 20 18:42:11 tema0 systemd-logind[1996]: Watching system buttons on /dev/input/event0 (Power Butt
on)
Sep 20 18:42:11 tema0 systemd-logind[1996]: Watching system buttons on /dev/input/event1 (Sleep Butt
on)
Sep 20 18:42:11 tema0 systemd-logind[1996]: Watching system buttons on /dev/input/event3 (Video Bus)
profe@tema0:/var/log$ _
```

5. EDICIÓN DE FICHEROS (touch, editor, vi)

El comando touch permite crear un fichero vacío. Con cualquier editor de texto se puede crear un fichero vacío pero con touch es especialmente cómodo y rápido.

```
profe@tema0:~$ ls
libros  musica  peliculas
profe@tema0:~$ cd libros
profe@tema0:~/libros$ cd ensayo
profe@tema0:~/libros/ensayo$ touch BorisVian.txt
profe@tema0:~/libros/ensayo$ ls
BorisVian.txt
profe@tema0:~/libros/ensayo$
```

El archivo se crea, pero vacío (es como mkdir para crear directorios).

Para escribir dentro de los archivos tenemos que hacer uso de algún editor de texto:

~ **editor** : es un editor de textos muy sencillo que podemos utilizar durante nuestras prácticas.

~ **nano**: es otro editor muy sencillo.

~ **vi**: Requiere cierto entrenamiento inicial para su uso, pero posibilita lograr una gran productividad en la edición de textos. Viene acompañado de un tutorial interactivo llamado vimtutor.

Dejo a tu elección el que quieras usar. En el hipotético caso de no estar instalado alguno de estos editores, su instalación es muy sencilla, basta con teclear **sudo apt-get install** seguido del nombre del programa que queremos instalar.

RESUMEN

- Todo usuario necesita un **nombre** y una **contraseña** para entrar en el sistema.
- La información se almacena físicamente en **directorios** y **subdirectorios** (carpetas y subcarpetas).
- Hay una serie de directorios predefinidos como /bin, /dev, /home, /etc, /var, etc. para todos los sistemas Linux.
- Hay **rutas absolutas,** que comienzan por el carácter "/", y que definen una ruta efectiva completa y **rutas relativas,** que no comienzan por el carácter "/", y cuya ruta efectiva sería la concatenación del directorio actual con esa misma ruta relativa. Los comandos vistos son los siguientes

EJERCICIOS (1ª parte)

En ocasiones, la respuesta a los ejercicios no se puede completar únicamente con el material teórico que se proporciona en este capítulo y el alumno debe, por tanto, buscar en otras fuentes complementarias.

En los ejercicios de este capítulo se recomienda consultar los manuales de los comandos mediante el comando **man**.

Para moverte por los directorios y archivos, es muy útil usar la tecla TABULADOR para autocompletar (pruébalo…a mi me cambió la vida).

Las respuestas para cada uno de estos ejercicios debe ir acompañada de una captura de pantalla donde se vea claramente que el resultado es el esperado.

1. Muestra el contenido del directorio /var/lib.
2. Muestra el contenido del directorio /var
3. Muestra el contenido del directorio /var/lib/apt/extended_states. ¿Se puede? ¿Por qué?
4. Muestra el contenido del archivo /etc/network/interfaces.
5. Vete a tu directorio personal. Muestra un listado del contenido de /usr/bin de las siguientes maneras:

a. Con una sola línea de comando.

b. Situándote en ese directorio y listando el contenido.

6. Muestra todos los archivos que hay en el directorio /var y dentro de sus directorios de manera recursiva en un solo comando.

7. Muestra todos los archivos que hay en el directorio /var con todas las características.

8. Muestra todos los archivos que hay en el directorio /var con todas sus características y ordenados por tamaño.

9. Ve al directorio /etc/kernel/postrm.d/ y muestra como una ruta absoluta el contenido del directorio raíz.

10. Crea en tu directorio personal un directorio que se llame 2DAW. Dentro de este directorio crea diferentes directorios para cada asignatura que tengas en el Ciclo. Dentro de cada una crea dos carpetas: una que se llame Teoria y otra que se llame Practica. Una vez hechas, muestra en forma de árbol la estructura que ha quedado dentro de tu directorio personal.

11. Crea dentro del directorio Teoria de la asginatura Despliegue de Aplicaciones Web un archivo llamado Tema0 y escribe dentro el texto "Las prácticas de Linux no son un coñazo y molan mogollón".

12. Vete al directorio /etc/network/ y desde allí muestra por pantalla el contenido del archivo Tema0 creado en el ejercicio anterior.

13. Modifica el archivo Tema0 y añade el siguiente texto:

"1.Linux mola
2.Java mola más
3.Los profes molan
4. Los alumnos también
5. Parece peloteo pero no lo es
6. Ya mismo voy a estar currando en una empresa
7. Arrierito somos y en el camino nos encontraremos
8. Al final me voy a enterar de como funciona la informática
9. Esta es la penúltima línea
10. Esta es la última línea
11. Te engañé…"

14. Muestra todo el contenido de Tema0.

15. Muestra las 4 primeras líneas del fichero Tema0.

16. Muestra las 2 últimas líneas del fichero Tema0.

17. Muestra todo el contenido del fichero Tema0 excepto la primera línea (se supone que no sabemos cuantas líneas tiene el fichero).

6. CARACTERES COMODÍN

Se pueden crear patrones usando **símbolos comodín** para no tener que escribir todos y cada uno de los ficheros.

Para mostrar cada uno de los ficheros que comienzan por fichero seguido de un número del uno al seis se puede utilizar un patrón:

```
profe@tema0:~$ ls
fichero1  fichero2  fichero3  fichero4  fichero5  libros  musica  peliculas
profe@tema0:~$ cat fichero[1-5]
contenido fichero 1
fin fichero 1
contenido fichero 2
fin fichero 2
contenido fichero 3
fin fichero 3
contenido fichero 4
fin fichero 4
contenido fichero 5
fin fichero 5
profe@tema0:~$ _
```

Si se quiere mostrar simplemente el contenido de todos los ficheros que comienzan por fichero se puede hacer:

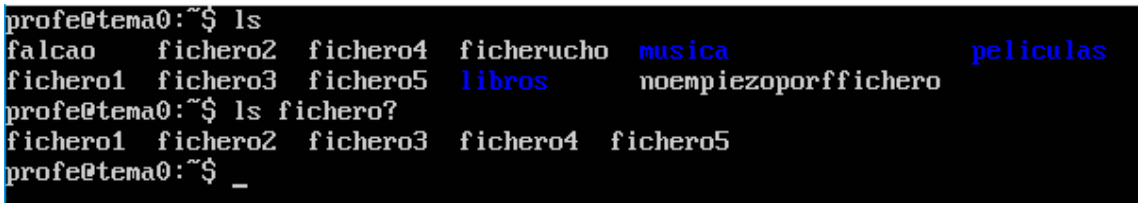
```
profe@tema0:~$ cat fichero*
contenido fichero 1
fin fichero 1
contenido fichero 2
fin fichero 2
contenido fichero 3
fin fichero 3
contenido fichero 4
fin fichero 4
contenido fichero 5
fin fichero 5
profe@tema0:~$ _
```

donde el carácter "*" representa cualquier combinación de caracteres, incluso la cadena vacía. Si existe un fichero con nombre fichero a secas en el directorio actual, también se mostrará.

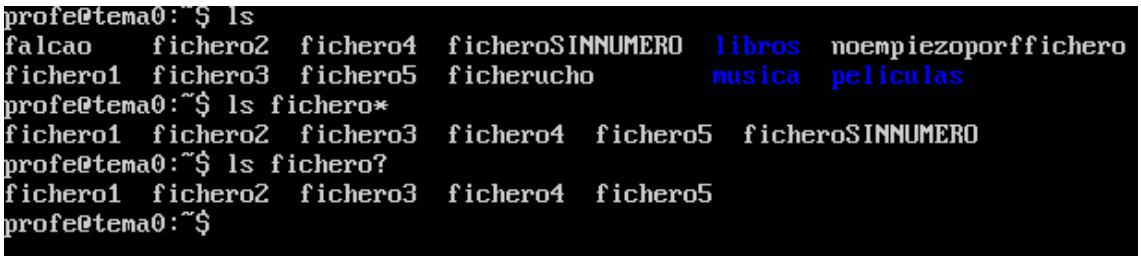
El carácter "*" se puede colocar en cualquier lugar. Por ejemplo, para mostrar todos los ficheros que empiezan por la letra f y terminan por la letra o dentro del directorio actual podemos usar el siguiente comando:

```
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficherucho  musica              peliculas
fichero1  fichero3  fichero5  libros      noempiezoporfichero
profe@tema0:~$ ls f*o
falcao  ficherucho
profe@tema0:~$ _
```

El símbolo "?" representa UN SOLO CARÁCTER. Por ejemplo, para mostrar todos los archivos que se llaman fichero + UN CARÁCTER CUALQUIERA podemos usar el siguiente comando:

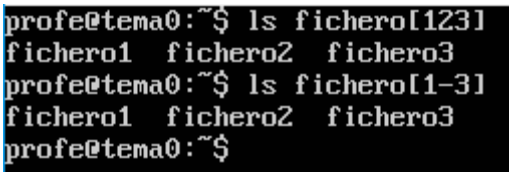


Si hubiéramos usado el *, y además de todos los archivos anteriormente creados, tenemos otro llamado ficheroSINNUMERO, nos lo hubiera listado también, aquí tenéis ambos ejemplos:



Los corchetes se utilizan de una forma parecida al carácter "?" aunque, a diferencia de éste, permiten afinar un poco más.

Por ejemplo [adfg] significa cualquiera de los caracteres a, d, f o g. [Hh]ola es un patrón que encaja tanto con Hola como con hola. [a-z]* representa cualquier cadena de caracteres que comienza con una letra minúsculas. Aquí tenéis dos ejemplos:



7. COPIAR, MOVER Y BORRAR FICHEROS Y DIRECTORIOS (cp, mv, rm, rmdir)

~ cp

El comando **cp** sirve para copiar ficheros. Se puede copiar un único fichero o muchos. Se pueden copiar tanto ficheros como directorios. Por supuesto, se pueden utilizar los símbolos comodín.

En el proceso de copia intervienen tres factores: lo que se copia, la ruta de origen y la ruta de destino. No está de más recordar que las rutas pueden ser tanto absolutas como relativas. La ruta de origen se especifica junto con lo que se quiere copiar.

Veamos como se copiaría el archivo hosts que está dentro de el directorio /etc/ a nuestro directorio actual (si no hubiéramos especificado ningún directorio de destino, también se hubiera copiado al directorio actual):

```
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  libros  noempiezoporffichero
fichero1  fichero3  fichero5  ficherucho        musica  peliculas
profe@tema0:~$ cp /etc/host
host.conf  hostname   hosts
profe@tema0:~$ cp /etc/hosts /home/profe/
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  hosts   musica                peliculas
fichero1  fichero3  fichero5  ficherucho        libros  noempiezoporffichero
profe@tema0:~$
```

Vamos a copiar todos los archivos fichero al directorio música usando el comodín *:

```
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  hosts   musica                peliculas
fichero1  fichero3  fichero5  ficherucho        libros  noempiezoporffichero
profe@tema0:~$ cp fichero* /home/profe/musica/
profe@tema0:~$ ls musica/
contemporanea  decada70  fichero1  fichero3  fichero5
decada60       decada80  fichero2  fichero4  ficheroSINNUMERO
profe@tema0:~$ _
```

Cuando se quiere especificar como directorio destino el directorio actual se puede utilizar también el carácter "."

- **mv**

El comando **mv** sirve para dos cosas, para mover y para cambiar de nombre. Se puede hacer cualquiera de las dos cosas por separado o las dos cosas al mismo tiempo:

Cambiar de nombre:

```
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  hosts   musica                peliculas
fichero1  fichero3  fichero5  ficherucho        libros  noempiezoporffichero
profe@tema0:~$ mv noempiezoporffichero siempiezoporffichero
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  hosts   musica     siempiezoporffichero
fichero1  fichero3  fichero5  ficherucho        libros  peliculas
profe@tema0:~$
```

Mover hosts de nuestro directorio al directorio libros

```
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  hosts   musica     siempiezoporffichero
fichero1  fichero3  fichero5  ficherucho        libros  peliculas
profe@tema0:~$ mv hosts libros/
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  libros  peliculas
fichero1  fichero3  fichero5  ficherucho        musica  siempiezoporffichero
profe@tema0:~$ ls libros/
ensayo  hosts  novela  poesia
profe@tema0:~$
```

Mover ficherucho a peliculas y además renombrarlo como ficheruchopeliculas:

```
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  libros  peliculas
fichero1  fichero3  fichero5  ficherucho        musica  siempiezoporffichero
profe@tema0:~$ mv ficherucho peliculas/ficheruchopeliculas
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  musica     siempiezoporffichero
fichero1  fichero3  fichero5  libros            peliculas
profe@tema0:~$ ls peliculas/
accion  animacion  clasicas  ficheruchopeliculas
profe@tema0:~$
```

- **rm**

El comando **rm** se utiliza para borrar ficheros. Es importante destacar que estos ficheros no se envían a una papelera así que NO SE PUEDEN RECUPERAR UNA VEZ BORRADOS.

Vamos a borrar el archivo ficheruchopeliculas del directorio películas:

```
profe@tema0:~$ cd peliculas/
profe@tema0:~/peliculas$ ls
accion  animacion  clasicas  ficheruchopeliculas
profe@tema0:~/peliculas$ rm ficheruchopeliculas
profe@tema0:~/peliculas$ ls
accion  animacion  clasicas
profe@tema0:~/peliculas$ _
```

- **COPIAR, MOVER Y BORRAR DIRECTORIOS**

De la misma manera que se copian, se borran o se mueven ficheros, se puede hacer lo mismo con los directorios.

Hay que tener en cuenta que un directorio puede contener muchos ficheros y, además, otros directorios que, a su vez, pueden contener más ficheros y directorios.

Por tanto, si se quiere copiar un fichero completo, con todo lo que tiene dentro, hay que indicarlo **con la opción -R**. A esto último se suele llamar "copiar de forma recursiva".

Por ejemplo, queremos pasar todo el contenido de libros a libros2. Primero probamos hacerlo directamente sin usar –R y no nos deja copiar el contenido, pero si usamos –R se copian todos los directorios y archivos que hay dentro del libros en libros2.

```
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  libros2  peliculas
fichero1  fichero3  fichero5  libros            musica   siempiezoporffichero
profe@tema0:~$ cp libros/* libros2/
cp: se omite el directorio 'libros/ensayo'
cp: se omite el directorio 'libros/novela'
cp: se omite el directorio 'libros/poesia'
profe@tema0:~$ cp -R libros/* libros2/
profe@tema0:~$ ls libros2/
ensayo  hosts  novela  poesia
profe@tema0:~$ _
```

El comando **mv** funciona de forma análoga a cp, pero mueve en lugar de copiar. Cuando se trata de renombrar, funciona exactamente igual que con los ficheros.

Ejemplo:

```
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  libros2  peliculas
fichero1  fichero3  fichero5  libros            musica   siempiezoporffichero
profe@tema0:~$ mv libros2 librosmalos
profe@tema0:~$ ls
falcao    fichero2  fichero4  ficheroSINNUMERO  librosmalos  peliculas
fichero1  fichero3  fichero5  libros            musica       siempiezoporffichero
profe@tema0:~$ _
```

Esto le cambia el nombre al directorio libros y pasa a llamarse librosmalos. El contenido de ese directorio permanece intacto.

Con **rm** se pueden borrar directorios pero hay que usar la forma recursiva, si no, no funciona:

```
profe@tema0:~$ rm librosmalos/
rm: no se puede borrar 'librosmalos/': Es un directorio
profe@tema0:~$ rm -R librosmalos/
profe@tema0:~$
```

Para los directorios también existen los comandos mkdir y rmdir (investígalos).

EJERCICIOS (2ª parte)

1. Muestra todos los ficheros del directorio /usr/bin que empiecen por g y acaben con e.
2. Muestra todos los archivos de /usr/bin que empiecen por k y tengan una a en la tercera posición.
3. Muestra los archivos del directorio /bin que terminen en n.
4. Crea un directorio en tu carpeta personal que se llame directorioPrueba. Copia todo el contenido del directorio /etc/dpkg en este nuevo directorio que acabas de crear.
5. Crea otro directorio llamado directorioPrueba2. Crea dentro de este directorio uno que se llame dpkg2 y haz que tenga el mismo contenido que dpkg.
6. Cambia el nombre de directorioPrueba2 a prueba2 y también cambia el nombre de directorioPrueba a prueba1
7. Crea un directorio que se llame todasLasPruebas y mueve las carpetas prueba1 y prueba2 a este directorio.
8. Crea el fichero archivoPrueba1.txt en el directorio prueba1 y escribe “Este es mi primer archivo de prueba”.

   Mueve el archivo a prueba2 y cambiale el nombre a archivoPrueba2.txt.

   Crea el fichero archivoPrueba2.txt en el directorio prueba1 y escribe “Este es mi segudno archivo de prueba”.

   Mueve todos los archivos de prueba1 a prueba2 de manera que en prueba2 se conserven los archivos que ya existían (si hay alguno con el mismo nombre, que no se machaque el antiguo).

8. ACTIVAR LA CUENTA DE SUPERUSUARIO

Como ya vimos anteriormente, para activar la cuenta "root" y acceder de esta manera directamente y vía la consola en modo "root", hay que teclear el siguiente comando:

$ sudo passwd root

Este comando permite activar la contraseña del usuario "root". Ésta será activada solo para las sesiones de la consola y no para la interfaz gráfica. Una vez que la contraseña haya comenzado, se podrá abrir una sesión en la consola como usuario "root".

$ sudo root

Esto nos permitirá, por ejemplo, **modificar los archivos de configuración que se encuentran dentro del directorio /etc (sólo un superusuario puede hacerlo**).

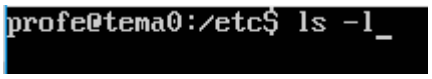
También podemos hacer modificaciones desde nuestro usuario usando delante del comando a utilizar el comando su

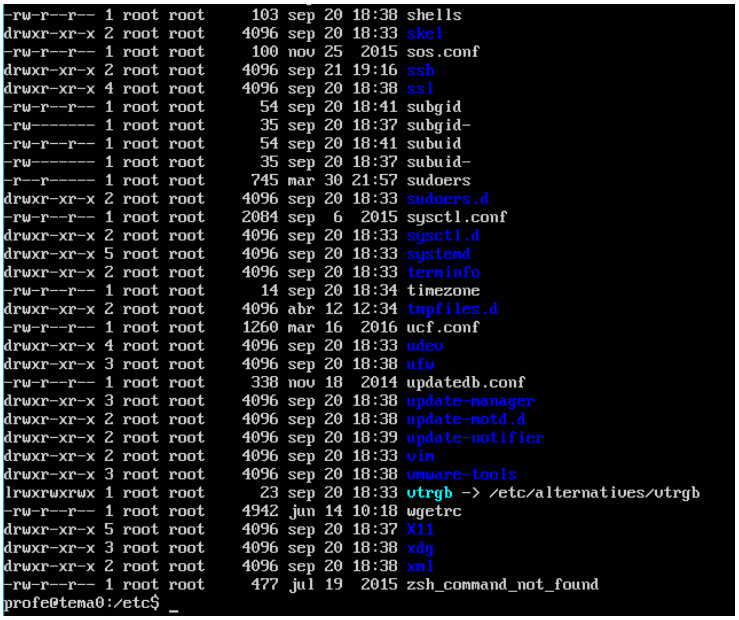
$ su editor /etc/hosts

Nos permitirá entrar en el archivo hosts y modificarlo. Si no lo hacemos así, podremos entrar, pero… ¡**nuestras modificaciones no se guardarán cuando salgamos del archivo**!

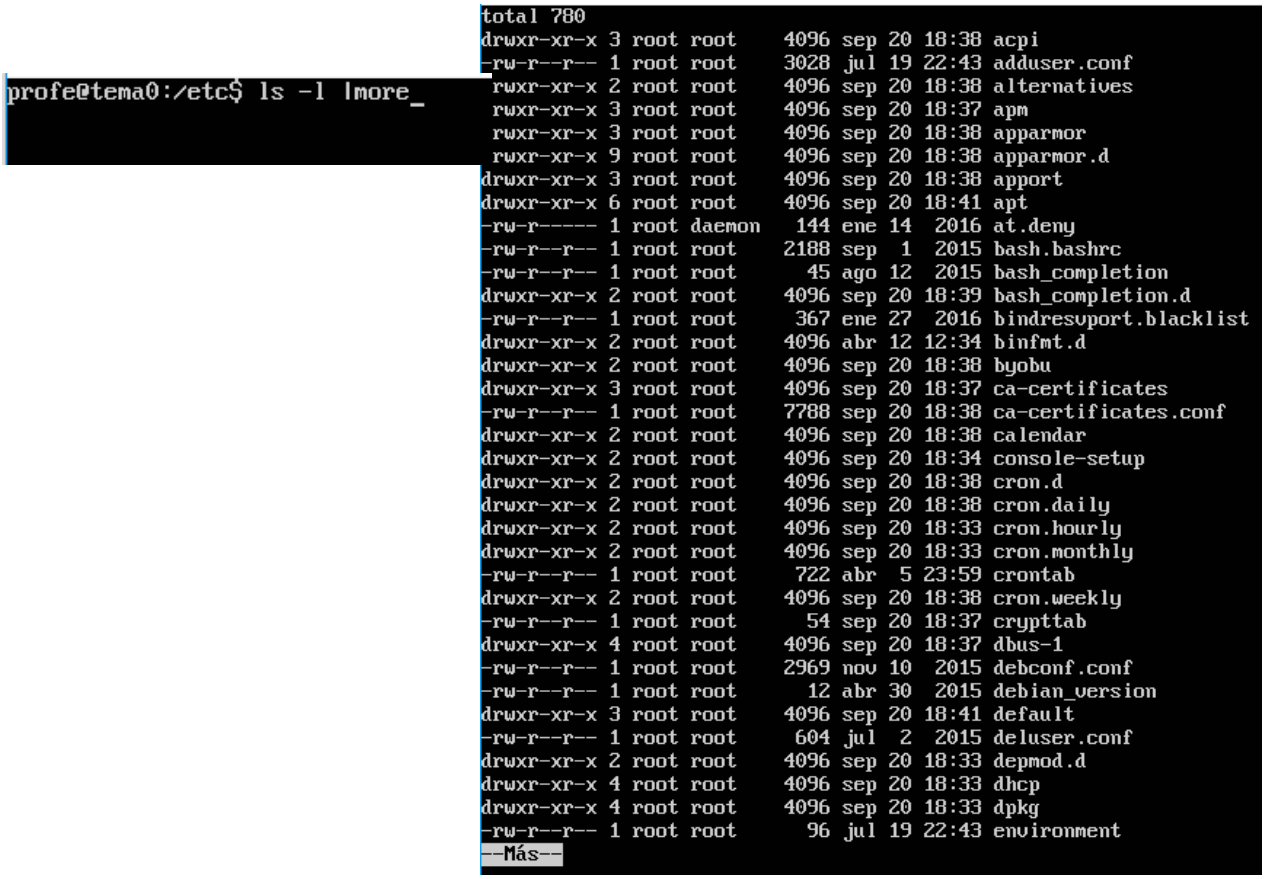
9. PERMISOS

La información sobre grupos, usuarios y permisos de los archivos se pueden obtener mediante el comando ls junto con la opción –l.





Si hay muchos archivos y queremos que se pare para ir viendo el listado, podemos usar la tubería |more. (el carácter "|" se consigue pulsando la tecla Alt+1).



Como vemos en estos archivos, en la primera columna aparecen los **permisos,** en la tercera se indica el **usuario** (en este caso es el administrador del sistema root) y en la cuarta columna aparece el nombre del **grupo** (que en estos casos es root excepto de at.deny que pertenece al grupo daemon).

Vamos a ver qué significan exactamente los caracteres de la primera columna:

-	r	w	x	r	-	x	r	-	x
Tipo de fichero.	Permisos para el dueño del fichero.			Permisos para el grupo al que pertenece el fichero.			Permisos para el resto de usuarios		

r	Permiso de **lectura.**
w	Permiso de **escritura.**
x	Permiso de **ejecución.**

El tipo de fichero se indica en la siguiente tabla (la d indica que es un directorio y no un archivo):

Tipo de fichero	
l	Enlace simbólico.
c	Dispositivo especial de caracteres.
b	Dispositivo especial de bloques.
P	FIFO (estructura de datos).
s	Socket (comunicaciones).
-	Ninguno de los anteriores. Puede ser un fichero de texto, un binario, etc.

Para el siguiente archivo:

Los permisos son de lectura y escritura para el propietario, pero no de ejecución (**rw-** r- - r- -).

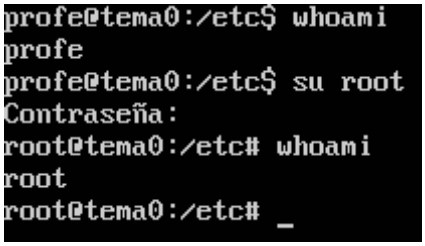
Los permisos son de lectura para los que pertenecen al grupo indicado (rw- **r- -** r- -).

Los permisos son de lectura para otros usuarios (rw- r- - **r- -**).

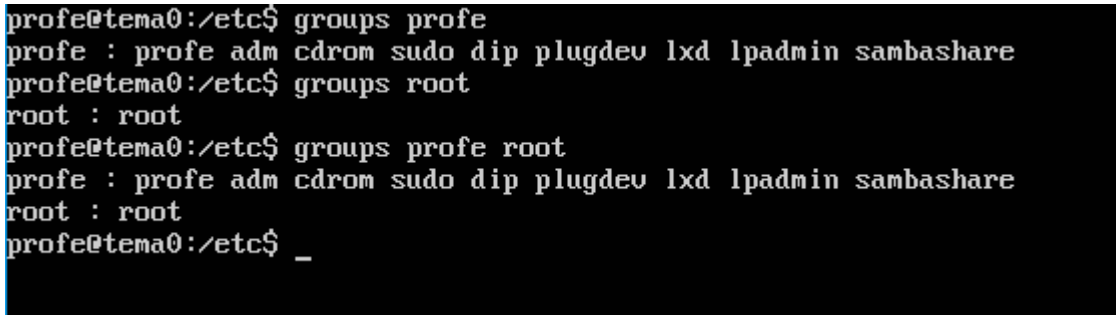
10. ¿QUIÉNES SOMOS? (whoami, groups)

Antes de empezar a crear usuarios, crear grupos y cambiar permisos, debemos saber quiénes somos y a qué grupo o grupos pertenecemos. Aunque, en principio, entremos en el sistema como un determinado usuario, podemos utilizar su para ejecutar comandos como otro usuario distinto, siempre y cuando sepamos la contraseña de ese otro usuario (como ya hemos visto con el usuario root).

Para saber que usuario somos se usa el comando **whoami**.



Para saber a que grupo pertenecemos se usa el comando groups.



11. GESTIÓN DE GRUPOS (groupadd, groupdel, groupmod)

Los comandos groupadd, groupdel y groupmod permiten crear, borrar y modificar grupos respectivamente.

Para usar estos comandos debemos hacerlo como superusuario. Vamos a crear el grupo iesvelazquez.

```
profe@tema0:/etc$ groupadd iesvelazquez
groupadd: Permission denied.
groupadd: cannot lock /etc/group; try again later.
profe@tema0:/etc$ sudo groupadd iesvelazquez
profe@tema0:/etc$
```

Para modificar el nombre del grupo usamos groupmod junto con el valor –n para darle un nuevo nombre, en este caso, cambiamos el grupo iesvelazquez por instituto_velazquez.

```
profe@tema0:/etc$ sudo groupmod -n instituto_velazquez iesvelazquez
profe@tema0:/etc$ _
```

Y para eliminar simplemente usamos groupdel

```
profe@tema0:/etc$ sudo groupdel instituto_velazquez
profe@tema0:/etc$ _
```

12. GESTIÓN DE USUARIOS (adduser, userdel, usermod)

La gestión de usuarios, al igual que la de grupos, exige que los comandos se ejecuten con los privilegios del administrador del sistema. Se puede escribir sudo antes de cada comando, o se puede hacer lo siguiente:

```
profe@tema0:/etc$ sudo bash
root@tema0:/etc#
```

Note el lector que el prompt ha cambiado. Ahora se muestra un carácter "#" en lugar de un "$". A partir de ahora, todos los comandos se ejecutarán con privilegios de administrador del sistema. Hay que acordarse de volver al usuario inicial mediante **exit**.

Tenemos dos grupos distitnos iesvelazquez e iesisbylia. Vamos a crear cuatro nuevos usuarios (profe1, profe2, profe3 y profe4). Los dos primeros los asignaremos al grupo iesvelazquez y el tercero lo asignaremos al grupo iesisbilya. El cuarto profe va a estar en los dos grupos.

```
root@tema0:/etc# adduser profe1 --ingroup iesvelazquez
Añadiendo el usuario `profe1' ...
Añadiendo el nuevo usuario `profe1' (1001) con grupo `iesvelazquez' ...
Creando el directorio personal `/home/profe1' ...
Copiando los ficheros desde `/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
Sorry, passwords do not match
passwd: Error de manipulación del testigo de autenticación
passwd: password unchanged
¿Intentar de nuevo? [s/N] s
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
Changing the user information for profe1
Enter the new value, or press ENTER for the default
        Full Name []: Profe
        Room Number []: Uno
        Work Phone []:
        Home Phone []:
        Other []:
¿Es correcta la información? [S/n] s
root@tema0:/etc# adduser profe2 --ingroup iesvelazquez
Añadiendo el usuario `profe2' ...
Añadiendo el nuevo usuario `profe2' (1002) con grupo `iesvelazquez' ...
Creando el directorio personal `/home/profe2' ...
Copiando los ficheros desde `/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
Changing the user information for profe2
Enter the new value, or press ENTER for the default
        Full Name []: Profe
        Room Number []: Dos
        Work Phone []:
        Home Phone []:
        Other []: _
```

```
Añadiendo el usuario `profe3' ...
Añadiendo el nuevo usuario `profe3' (1003) con grupo `iesisbilya' ...
Creando el directorio personal `/home/profe3' ...
Copiando los ficheros desde `/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
Changing the user information for profe3
Enter the new value, or press ENTER for the default
        Full Name []: Profe
        Room Number []: Tres
        Work Phone []:
        Home Phone []:
        Other []:
¿Es correcta la información? [S/n] s
root@tema0:/etc# adduser profe4 --ingroup iesisbilya
Añadiendo el usuario `profe4' ...
Añadiendo el nuevo usuario `profe4' (1004) con grupo `iesisbilya' ...
Creando el directorio personal `/home/profe4' ...
Copiando los ficheros desde `/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
Changing the user information for profe4
Enter the new value, or press ENTER for the default
        Full Name []: Profe
        Room Number []: 4
        Work Phone []:
        Home Phone []:
        Other []:
¿Es correcta la información? [S/n] s
root@tema0:/etc# groups profe1 profe2 profe3 profe4
profe1 : iesvelazquez
profe2 : iesvelazquez
profe3 : iesisbilya
profe4 : iesisbilya
root@tema0:/etc#
```

Nos falta añadir profe4 al grupo iesvelazquez:

```
root@tema0:/etc# adduser profe4 iesvelazquez
Añadiendo al usuario `profe4' al grupo `iesvelazquez' ...
Adding user profe4 to group iesvelazquez
Hecho.
root@tema0:/etc# groups profe1 profe2 profe3 profe4
profe1 : iesvelazquez
profe2 : iesvelazquez
profe3 : iesisbilya
profe4 : iesisbilya iesvelazquez
root@tema0:/etc#
```

Al crear los usuarios, se nos han pedido las claves, no obstante estas claves se pueden cambiar con el comando passwd.

```
# passwd profe1
# passwd profe2
# passwd profe3
```

Recuerde el lector salir del modo root con el comando exit cuando no tenga que hacer tareas que requieran privilegios de administrador.

```
# exit
```

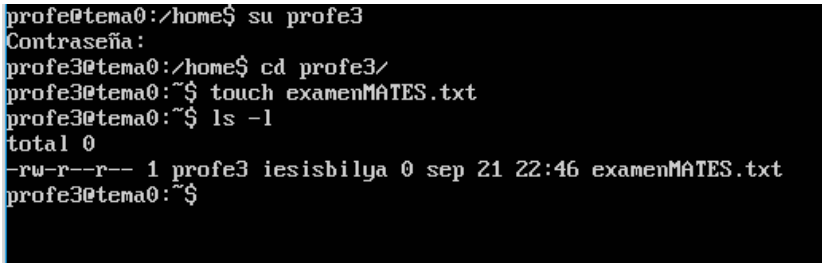
De ahora en adelante, simplemente se indicará con el carácter "$ " que se trabaja como usuario sin privilegios y con el carácter "#" que se trabaja como root.

Cabe señalar que para cada usuario, se crea por defecto un directorio dentro de /home. Cuando un usuario se conecta al sistema, "aterriza" en ese directorio. Es lo que hemos denominado anteriormente como el directorio de trabajo.

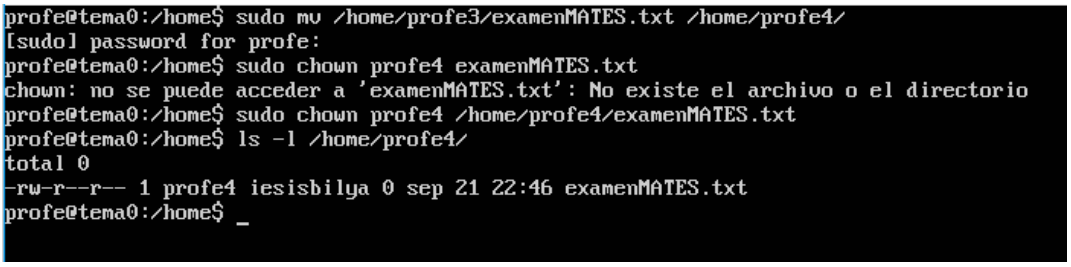
```
root@tema0:/etc# exit
exit
profe@tema0:/etc$ cd /home/
profe@tema0:/home$ ls
profe  profe1  profe2  profe3  profe4
profe@tema0:/home$
```

13. CAMBIO DE GRUPO Y DE DUEÑO (chown, chgrp)

Vamos a crearnos un archivo llamado examenMATES.txt con nuestro usuario profe3 (tiene que ser dentro de nuestra carpeta /home/profe3, si no, no tendremos permiso).



Todo esto se puede cambiar. Moveremos el fichero al directorio de trabajo del usuario profe4 y le cambiaremos el dueño.



Para poder hacer estos cambios, tenemos que estar o en usuario root o en un usuario que sea suoders (debe ser modificado en un fichero, pero ahora no toca ver eso) como profe, en mi caso (ni profe1, profe2, profe3, profe4 lo son).

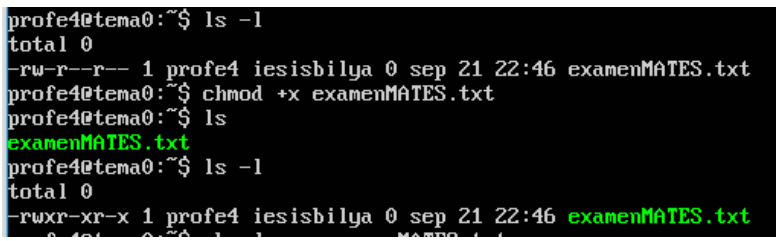
Como vemos, ahora el archivo está en la carpeta de profe4 y además es el propietario.

Tanto chown como chgrp se pueden usar con la opción -R para cambiar el dueño o el grupo en un directorio completo, de forma recursiva.

14. CAMBIO DE PRIVILEGIOS (chmod)

El comando chmod sirve para cambiar los permisos de uno o varios ficheros. Esos mismos permisos que se pueden ver con ls –l.

Vamos a cambiar los permisos para examenMATES.txt desde el usuario profe4.



Hemos dado permisos de ejecución a todos los usuarios (propio usuario, los del grupo, otros).

Ahora vamos a quitarle los permisos de ejecución a los usuarios del grupo:

```
profe4@tema0:~$ chmod g-x examenMATES.txt
profe4@tema0:~$ ls -l
total 0
-rwxr--r-x 1 profe4 iesisbilya 0 sep 21 22:46 examenMATES.txt
```

Y ahora vamos a quitarle los permisos de lectura y ejecución a otros usuarios:

```
profe4@tema0:~$ chmod o-r-x examenMATES.txt
profe4@tema0:~$ ls -l
total 0
-rwxr----- 1 profe4 iesisbilya 0 sep 21 22:46 examenMATES.txt
profe4@tema0:~$
```

Cuando no se especifica ninguna de estas tres letras correspondientes a los usuarios (u, g, o) como en el ejemplo anterior, se sobreentiende que nos referimos a todos ellos. Se puede indicar de forma explícita con el carácter a (all).

Para entenderlo mejor, en la siguiente tabla, se muestran de forma esquemática, los parámetros del comando chmod:

u	g	o	+	r	w	x
(user) dueño del fichero	(group) usuarios que pertenecen al mismo grupo	(others) el resto de usuarios	dar permiso quitar permiso	(read) lectura	(write) escritura	(execution) ejecución

A este método, que utiliza los caracteres rwx se le denomina método simbólico. Podemos utilizar de forma análoga el método numérico, usando notación octal

r	**w**	**x**	1	1	1	= **7**
r	**w**	**-**	1	1	0	= **6**
r	**-**	**x**	1	0	1	= **5**
r	**-**	**-**	1	0	0	= **4**
-	**w**	**x**	0	1	1	= **3**
-	**w**	**-**	0	1	0	= **2**
-	**-**	**x**	0	0	1	= **1**

Así, para asignarle al usuario todos los permisos (r w x valen 1 1 1, en octal 7), a usuarios del grupo sólo permisos de lectura y ejecución (r w x valen 1 0 1, en octal 5) y a otros sólo permisos de lectura y ejecución también (r w x valen 1 0 1, en octal 5) lo haremos así:

```
profe4@tema0:~$ ls -l
total 0
-rwxr----- 1 profe4 iesisbilya 0 sep 21 22:46 examenMATES.txt
profe4@tema0:~$ chmod 755 examenMATES.txt
profe4@tema0:~$ ls -l
total 0
-rwxr-xr-x 1 profe4 iesisbilya 0 sep 21 22:46 examenMATES.txt
profe4@tema0:~$ _
```

Los permisos de los directorios se pueden cambiar de la misma forma que los ficheros, aunque el significado es algo diferente. Si un directorio tiene el permiso de lectura quiere decir que se puede ver su contenido. Si tiene permiso de escritura, quiere decir que se pueden crear ficheros dentro y si tiene permiso de ejecución quiere decir que se puede entrar dentro.

EJERCICIOS (3ª PARTE)

1. Completa la siguiente tabla:

210	
	r-xrw-r-
	rw—wxrw-
117	
543	
	r- -r- - r w x

2. Crea los grupos 1Ciclo y 2Ciclo.

3. Crea los usuarios menganito y zutanito y añádelos a 1Ciclo

4. Crea los usuarios margarita y juanita y añádelos a 2Ciclo.

5. Como usuario menganito, crea un archivo evaluación_inicial.txt al que solamente él tenga acceso tanto de lectura como de escritura.

6. Crea como usuario margarita un archivo trabajoMATES.txt en el que tenga todos los permisos, y que todos los de su mismos grupo tengan permisos de lectura y escritura. Comprueba como usuaria juanita que puedes modificar el fichero.

7. Como usuario zutanito, crea un archivo apuntesLENGUA.txt que pueda ser leído por cualquier usuario y de su grupo cualquiera pueda leer y escribir.

8. Crea el usuario fulanito que pertenezca a 2Ciclo. Crea una carpeta llamada compartida_con_todos y dentro de ella el archivo tema1.txt.

9. Da permiso de lectura para que todo el mundo pueda leer de la carpeta compartida_con_todos así como de todos los ficheros que hay dentro.

10. Si un usuario tiene permiso de lectura sobre un fichero, pero ese fichero se encuentra dentro de un directorio sobre el que no tiene permiso de lectura ¿podrá leer el fichero?, haz la prueba.