

CICLO FORMATIVO DE GRADO SUPERIOR - TÉCNICO
EN ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN
RED

IMPLANTACIÓN DE SISTEMAS OPERATIVOS

LINUX

Ficheros y directorios



Índice

1.1.- ENTRADA AL SISTEMA	2
1.2.- ESTRUCTURA DE DIRECTORIOS	2
1.3.- VISUALIZACIÓN, CREACIÓN Y CAMBIO DE DIRECTORIO	3
1.4.- VISUALIZACIÓN DE FICHEROS	6
1.5.- EDICIÓN DE FICHEROS	7
1.5.- CARACTERES COMODÍN	9
1.6.- COPIA Y BORRADO DE FICHEROS	10
1.7.- COPIA Y BORRADO DE DIRECTORIOS	12
RESUMEN	14

1.1.- ENTRADA AL SISTEMA

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:~\$

El prompt indica el nombre de usuario y el nombre de la máquina, 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”, visto como ejemplo en el capítulo anterior.

1.2.- ESTRUCTURA DE DIRECTORIOS

En un ordenador, el almacenamiento de información se lleva a cabo mediante carpetas y subcarpetas. 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:

/ directorio raíz	/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.

1.3.- VISUALIZACIÓN, CREACIÓN Y CAMBIO DE DIRECTORIO (pwd, ls, cd, mkdir)

Comando 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 perdidos.

```
nira@ubuntu-desktop:~$ pwd
```

```
/home/nira
```

Comando 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.

```
nira@ubuntu-desktop:~$ ls
```

```
Desktop Documents Example Music Pictures Public
Templates Videos
```

Se pueden añadir opciones a ls:

```
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 -lh
```

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:

```
nira@ubuntu-desktop:~$ man ls
```

(Para más información, añade el comando y posteriormente --help. Ej. ls --help)

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

Comando 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. Recordemos que el comando cd sin argumentos, nos lleva al directorio de trabajo personal.

Comando mkdir

Se pueden crear directorios con el comando mkdir. Por ejemplo, para crear una estructura de carpetas donde un estudiante guardará información sobre sus asignaturas según el siguiente esquema:

/home/nira	/Documentos			
	/Escritorio			
	/Imágenes			
	/Música			
	/asir	/curso_01	/iso	/teoria /actividades
			/hardware	/teoria
	/Video			

Para crear estas carpetas y subcarpetas se haría lo siguiente:

```
~$ mkdir asir
~$ cd asir/
~/asir$ mkdir curso_01
~/asir$ cd curso_01/
~/asir/curso_01$ mkdir iso hardware
~/asir/curso_01$ ls
iso  hardware
~/asir/curso_01$ cd iso/
~/asir/curso_01/iso$ mkdir teoria actividades
~/asir/curso_01/iso$ cd ..
~/asir/curso_01$ cd hardware
~/asir/curso_01/hardware$ mkdir teoria
```

Nótese que ya no se muestra en el ejemplo el prompt completo, con el nombre de usuario y el nombre de la máquina. Se seguirá en el libro esta norma a partir de ahora.

1.4.- 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.

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:

```
~$ cat doc1.txt
```

Muestra el contenido del archivo documento1. Si el lector ha probado a hacerlo con un contenido muy extenso, se habrá dado cuenta de que es imposible ver todo el contenido de este fichero, porque ha pasado por pantalla muy rápido. Por eso `cat` se suele utilizar para visualizar el contenido de archivos pequeños.

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:

```
~$ more doc1.txt
```

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”:

```
:~$ less doc1.txt
```

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

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). Por defecto, tanto `head` como `tail` muestran 10 líneas, pero eso se puede cambiar con la opción `-n`.

```
:~$ head doc1.txt
```

```
:~$ tail doc1.txt
```

1.5.- EDICIÓN DE FICHEROS (touch, vi, ee, mcedit)

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.

```
~$ ls
```

```
Descargas  Documentos  Imágenes  Plantillas  Escritorio....  
Musica....Videos
```

```
~$ touch prueba.txt
```

```
~$ ls
```

```
Descargas  Documentos  Imágenes  Plantillas  Escritorio  
Musica  Videos  prueba.txt
```

```
~$ cat prueba.txt
```

```
~$
```

Se puede ver en el ejemplo cómo se ha creado el archivo `prueba.txt` pero al visualizar su contenido con `cat`, no aparece nada en pantalla, por tanto está vacío.

El programa `ee` es un editor muy rudimentario, pero al mismo tiempo efectivo. Podemos editar el archivo anterior y escribir alguna frase:

```
~$ ee prueba.txt
```

Presionando la tecla `ESC`, el usuario puede salir al menú principal y guardar el fichero. Podemos comprobar ahora cuál es el contenido del fichero:

```
~$ cat prueba.txt
```

```
Hola, aquí estoy aprendiendo Linux.
```

Otro editor muy simple es `nano`. Se deja al lector curioso probar su funcionamiento y compararlo con `ee`. En el 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.

Por ejemplo, si queremos instalar `ee`:

```
~$ sudo apt-get install ee
```

El programa `mcedit` es un editor algo más sofisticado que `ee` o `nano` (al menos en apariencia) y es una parte de `mc` (Midnight Commander), un programa muy al estilo del famoso Norton Commander de MS-DOS. Vamos a modificar el archivo `prueba.txt` creado anteriormente. Antes de eso, instalaremos `mc`, ya que no está instalado por defecto:

```
~$ sudo apt-get install mc
```

```
~$ mcedit prueba.txt
```

Con la tecla `F2` guardamos los cambios y con dos pulsaciones de `ESC` (o con la tecla `F10`) salimos del programa.

Comprobamos ahora que todo se ha grabado bien:

```
~$ cat prueba.txt
```

```
Hola, aquí estoy aprendiendo Linux.
```

```
A ver qué tal se me da.
```

Hemos dejado para el final al editor de Linux por excelencia, se trata de `vi`.

A primera vista es el más difícil de utilizar, lo cual es cierto, y parece que tiene menos opciones, pero se trata de un programa muy potente. Cualquier “linuxero” que se precie sabe manejar bien este programa.

Añadiremos una línea más al fichero `prueba.txt`. Para ello, seguiremos los siguientes pasos:

```
~$ vi prueba.txt
```

- Pulsar la letra “`i`” para entrar en modo “edición”.
- Escribiremos el texto.
- Pulsar la tecla `ESC` para salir del modo “edición” o `ctrl+C`

- Teclear “:” + “w” + INTRO para grabar los cambios.
- Teclear “:” + “q” + INTRO para salir del programa.

Comprobamos una vez más que todo está bien grabado:

```
~$ cat prueba.txt
```

```
Hola, aquí estoy aprendiendo Linux.
```

```
A ver qué tal se me da.
```

```
Por ahora parece bastante sencillo.
```

Es más que recomendable realizar el tutorial llamado vimtutor.

1.6.- CARACTERES COMODÍN

En muchas ocasiones es necesario realizar acciones sobre muchos archivos o directorios al mismo tiempo. Por ejemplo:

```
$ cat docu1 docu2 docu3 docu4 docu5 docu6
```

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

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

```
$ cat docu[1-6]
```

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

```
$ cat docu*
```

donde el carácter “*” representa cualquier combinación de caracteres, incluso la cadena vacía. Si existe un fichero con nombre docu 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 a y terminan por la letra s dentro del directorio /usr/bin:

```
$ ls /usr/bin/a*s
```

El símbolo “?” representa un carácter cualquiera. De esta forma, la siguiente sentencia muestra todos los ficheros del directorio /usr/bin cuyo nombre comienza por g, sigue cualquier carácter, a continuación, sigue una o y termina con cualquier cadena de caracteres incluida la cadena vacía:

```
$ ls /usr/bin/g?o*
```

Los **corchetes** se utilizan de una forma parecida al carácter “?” aunque, a diferencia de éste, permiten especificar 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.

1.7.- COPIA Y BORRADO DE FICHEROS (cp, mv, rm)

Comando 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. La ruta de origen se especifica junto con lo que se quiere copiar. Veamos un ejemplo:

```
$ cp /etc/hosts /home/alumno/pruebas/
```

La sentencia anterior copia el fichero hosts, que se encuentra en el directorio /etc al directorio /home/ alumno/pruebas/.

Si no se especifica ningún directorio origen, se toma por defecto el directorio actual.

Por ejemplo:

```
$ cp *.odt textos/
```

copia todos los archivos con la extensión odt del directorio actual al directorio textos. Cuando se quiere especificar como directorio destino el directorio actual se utiliza el carácter “.” Por ejemplo:

```
$ cp /usr/bin/g* .
```

copia todos los ficheros del directorio /usr/bin que comienzan por la letra g al directorio actual.

Comando 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. Por ejemplo:

```
$ mv mi_texto.txt carta.txt
```

le cambia el nombre a mi_texto.txt y pasa a llamarse carta.txt. En cambio:

```
$ mv carta.txt Documentos/
```

mueve carta.txt al directorio Documentos.

Se pueden hacer las dos cosas a la vez, mover y cambiar el nombre:

```
~$ cd Documentos/
```

```
~/Documentos$ mkdir correspondencia
```

```
~/Documentos$ mv carta.txt correspondencia/carta01.txt
```

En este caso, el fichero carta.txt se ha movido al directorio ~/Documentos/correspondencia y además se le ha cambiado el nombre a carta01.txt

Comando 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**.

Ejemplo:

```
$ rm *.txt
```

Esta sentencia borra todos los archivos con la extensión txt del directorio actual.

1.8.- COPIA Y BORRADO DE DIRECTORIOS (cp, mv, rm)

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”**.

Ejemplo:

```
~$ mkdir multimedia2
```

```
~$ cp multimedia/* multimedia2
```

```
~$ ls multimedia2
```

Se ha hecho una copia del contenido del directorio multimedia al directorio multimedia2, pero no se ha copiado ningún archivo ¿qué ha pasado?. Sencillamente no se ha hecho una copia recursiva (con la opción -R). Se ha intentado copiar únicamente justo dentro del directorio multimedia pero no a un nivel inferior. Como a ese nivel no había ningún fichero, no se ha copiado nada.

Vamos a intentarlo ahora de forma recursiva:

```
~$ cp -R multimedia/* multimedia2
```

```
~$ ls -R multimedia2 multimedia2:
```

```
imagenes    musica    presentaciones    video
```

```
multimedia2/imagenes:
```

```
otras personales multimedia2/imagenes/otras: multimedia2/imagenes/personales:
```

```
multimedia2/musica:
```

```
estilos_favoritos.txt multimedia2/presentaciones:
```

```
multimedia2/video:
```

Como se puede comprobar, se han copiado tanto la estructura de directorios como los contenidos de cada uno de ellos.

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:

```
~$ mv multimedia2 multimedia_copia
```

Esto le cambia el nombre al directorio multimedia2 y pasa a llamarse multimedia_copia. El lector puede comprobar que el contenido de ese directorio permanece intacto.

Con **rm** se pueden borrar directorios.

```
~$ rm multimedia_copia/
```

rm: no se puede borrar «multimedia_copia/»: Es un directorio

Se obtiene un error, ¿qué sucede? Que hay que borrar el contenido de la siguiente forma:

```
~$ rm -Rf multimedia_copia/
```

Además de la opción -R, se ha incluido la opción -f que hace que no se nos pida confirmación por cada elemento que se quiere borrar.

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.

- **Utilización de los símbolos comodín:**

Ejemplos	Significado
----------	-------------

*	Cualquier cadena de caracteres.
---	---------------------------------

f	Cadena de caracteres que contienen una f.
-----	-------------------------------------------

z*	Cadena de caracteres que empieza por z y le sigue cualquier cosa.
----	-------------------------------------------------------------------

a?	Una cadena formada por dos caracteres, el primero una a y el segundo, cualquier carácter.
----	-------------------------------------------------------------------------------------------

[Dd]ocumento Puede ser Documento o documento.

A[a-z][0-6]	Una cadena formada por la A mayúscula seguida de cualquier letra minúscula, seguida a su vez de un dígito del 0 al 6.
-------------	-----------------------------------------------------------------------------------------------------------------------

Los comandos vistos en este capítulo son los siguientes:

COMANDO	ACCIÓN
pwd	Muestra el directorio actual
ls	Lista ficheros y directorios
cd	Cambia de directorio
mkdir	Crea uno o varios directorios
cat	Visualiza un fichero
more	Visualiza un fichero pantalla a pantalla
less	Visualiza un fichero pantalla a pantalla y permite retroceder
head	Visualiza las primeras filas de un fichero
tail	Visualiza las últimas filas de un fichero
touch	Crea un fichero vacío
ee	Editor de textos muy simple
mcedit	Editor de textos que forma parte de MidnightCommander
vi	Editor de textos muy potente
apt-get	Instala y desinstala programas
man	Muestra ayuda sobre un determinado comando
cp	Copia archivos o directorios
mv	Mueve o renombra archivos o directorios
rm / rm-R	Borra archivos /Borra directorios