

## CONCEPTOS BÁSICOS

### ¿Que es el Prompt?

Pues no es más que la línea desde donde el sistema nos indica que está listo para recibir órdenes. Generalmente se parece a algo como:

```
[usuario/ dominio directorio]$
```

En caso de estar autenticado como superusuario (root) suele parecerse a:

```
[usuario/ dominio directorio]#
```

### El SHELL

El programa que se ejecuta en Linux cada vez que un usuario se conecta y que le permite interactuar con el sistema se conoce como **shell** (bash normalmente). Este es capaz de interpretar una gran cantidad de comandos y sentencias.

Nosotros trabajaremos con el shell bash (**Bourne Again Shell**), que es un shell o intérprete de comandos GNU que incorpora la mayoría de distribuciones de Linux. Compatible con el shell sh.

Algo muy útil es la función **auto completar durante la escritura**. Al teclear uno o varios caracteres se puede pulsar TAB para completar de forma unívoca un comando, nombre de fichero o una variable.

### Obtener ayuda en el terminal Linux

Existen múltiples y variadas formas de obtener ayuda en un sistema Linux. Veamos algunas de ellas:

1. Muchos comandos poseen una opción para mostrar una ayuda breve acerca de su utilización. Esta opción usualmente es **-h**, **-help** ó **-?**.

```
chmod --help          // ayuda para el comando chmod.
```

2. **man** manual en línea bastante amplio acerca de comandos, formatos de ficheros de configuración, llamados al sistema, etc.

```
man chmod             // manual del comando chmod
```

3. **info** despliega información acerca de comandos, en ocasiones más amplia que la que brinda man.

```
info ln               // información sobre el comando ln
```

## USUARIOS Y GRUPOS

El usuario con más privilegios en Linux es **root**. Es el único con derechos suficientes para crear o eliminar a otros usuarios, además de acceder a todo el sistema de ficheros sin ninguna restricción.

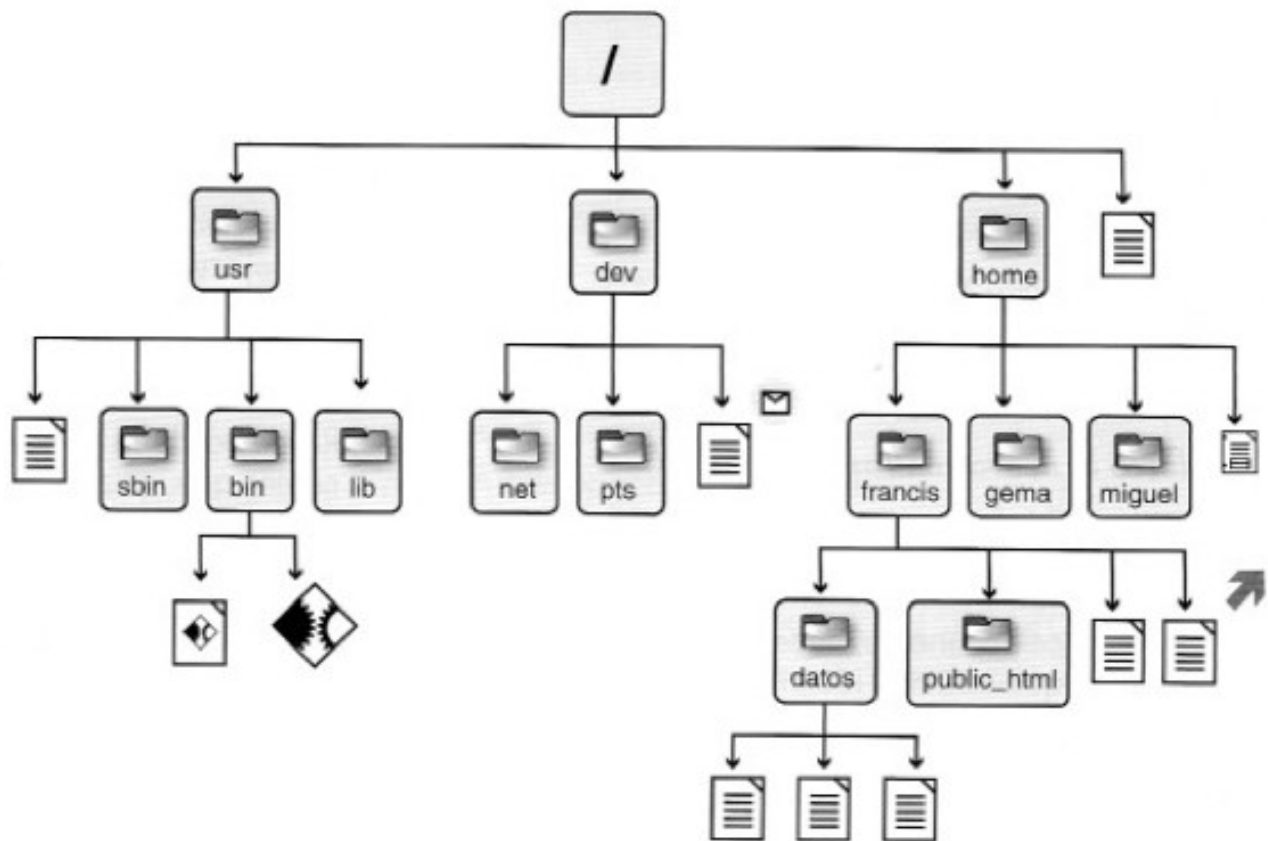
## Comandos su y sudo

**su** son las siglas de «Switch User» y sirve para cambiar de usuario sin necesidad de cerrar o cambiar de sesión.

`su jose` // cambia al usuario jose

`su -` // cambiar a root dentro de su carpeta personal y demás variables de entorno.

## FICHEROS Y DIRECTORIOS



## Comando ls

Muestra un listado del contenido de un directorio.

Sintaxis: `ls [opciones] [directorio | fichero]`

### Algunas opciones

- l Muestra la salida en formato largo.
  - r Lista el contenido de un directorio en orden inverso.
  - a Lista los ficheros ocultos (sus nombres comienzan con punto).
  - lh Muestra el tamaño de los ficheros en forma más legible (16M, 4k, etc.)
- `ls -la /etc`

### Comando cd

Se utiliza para cambiar de directorios y moverse por el sistema de archivos.

Sintaxis: `cd [directorio]`

```
cd /tmp      // cambia al directorio tmp.
cd           // cambia al directorio home del usuario.
cd ~         // cambia al directorio home del usuario.
cd ..        // retrocede al directorio anterior.
cd ../..     // retrocede 2 directorios.
```

### Comando pwd

Indica el camino absoluto del directorio en el cual nos encontramos actualmente.

### Comando mkdir

Sirve para crear nuevos directorios.

Sintaxis: `mkdir [opciones] [nombre_directorio]`

```
mkdir bin    // Crea un directorio con el nombre bin en el directorio actual
mkdir /bin   // Crea un directorio con nombre bin en la raíz.
```

### Comando mv

**mv** permite mover o renombrar ficheros o directorios.

```
mv *.txt /tmp           // Mueve ficheros terminados en .txt al
directorio /tmp.
```

### Comando cp

Sirve para copiar un fichero en otro, o varios ficheros en un directorio.

Sintaxis:

```
cp [opciones] <fuente> <destino>
cp [opciones] <ficheros> <directorio>
```

(Crea acceso directo).

### Algunas opciones

-R Copia recursivamente un directorio. (copia todo lo que cuelga del directorio).

```
cp /documentos/fich.txt . // copia el fichero fich.txt en el directorio actual
```

### Comando rm

Se utiliza para borrar y/o desenlazar ficheros.

Sintaxis: `rm [opciones] [ficheros | directorios]`

### Algunas opciones

-r Borra recursivamente un directorio (borra todo lo que cuelga del directorio).

-f Borra forzosamente en caso de que no se tenga permiso de escritura en forma directa.  
`rm prueba` // Borra el fichero prueba.  
`rm -rf temp/` // Fuerza el borrado del directorio temp y todo su contenido.  
  
`chgrp ftp_user /usr/ftp` // Cambia el grupo a ftp\_user para el directorio ftp.

### Comando diff

Nos permite comparar ficheros de texto línea a línea, y nos indica en caso de que no sean iguales en que líneas cambian.

Sintaxis: `diff [opciones] <directorio1 | fichero1> <directorio2 | fichero2>`

`diff fichero1 fichero2`

### Comando cat

Concatena ficheros y los imprime en la salida estándar, normalmente la pantalla. Si no se le pasa ningún argumento muestra el contenido del fichero. E

Sintaxis: `cat [opciones] [fichero]`

Igual que >, pero la salida de la orden se añade a la salida que indiquemos. (no reemplaza el contenido del fichero destino).

### Algunas opciones

- > Redirecciona stdout. Vaciar o volcar a un destino no sea CON (monitor), (si el destino contiene algo lo sobrescribe).
- 2> Redirecciona stderr. Es decir, nos permite volcar una salida de error en un destino que no sea CON (monitor).
- < Redirecciona stdin. Es decir, nos permite indicar una entrada para la orden que no sea CON (teclado).
- >>

| El indicador de tubería. Nos permite indicar que la entrada de una orden será la salida de otra orden. Es decir, el stdout de la 1ª orden, será el stdin de la 2ª orden.

P? Localiza ficheros que contengan una 'P' y otra letra más.

P\* Localiza ficheros que empiezan con 'P'.

```
cat /etc/passwd           // Lista el contenido del fichero passwd.
cat > fichero              //Crea fichero y permite editarlo (si el fichero existe
                             lo reemplaza).
cat < origen > destino     // Copia origen en destino.
cat fichero1 fichero2 > fichero3 // Concatena el contenido de los ficheros
fichero1 y fichero2 en el fichero fichero3.
cat /dev/null > fichero    // Vacía el contenido del fichero sin borrarlo.
```

### Comandos more y less

Paginan uno o varios ficheros y los muestran en pantalla. El comando **more** (salta en páginas) y **less** (salta en líneas).

**More** muestra sucesivamente el porcentaje del fichero visto hasta el momento aunque es más restrictivo en cuanto al movimiento dentro del texto, mientras que **less** no limita este aspecto pues acepta el empleo de todas las teclas de movimiento tradicionales.

Sintaxis:

```
more [fichero]
```

```
less [fichero]
```

### Algunas teclas que podemos usar

q Permite interrumpir el proceso y salir.

```
less /etc/passwd
```

```
more /etc/passwd
```

```
cat fichero | less
```

### Comandos tail y head

Muestran respectivamente el final y el comienzo (10 líneas por defecto) de uno o varios ficheros. De no especificarse al menos un fichero toman la entrada estándar.

Sintaxis:

```
tail [opciones] [ficheros]
```

```
head [opciones] [ficheros]
```

Imprime las n últimas/primeras líneas en lugar de las diez establecidas por defecto.

### Algunas opciones

-<n>

```
tail -20 /var/log/secure // Imprime las ultimas 20 líneas del fichero
secure.
```

```
head -10 /var/spool/mail/jose // Imprime las 10 primeras líneas del fichero
jose.
```

### Comando wc

Imprime el número de líneas, palabras y bytes de uno o varios ficheros. Si son varios ficheros hace también un resumen de los totales.

Sintaxis: wc [opciones] [ficheros]

### Algunas opciones

-l Sólo cuenta líneas.

-c Sólo cuenta bytes.

-w Sólo cuenta palabras.

```
wc -l /etc/passwd
```

```
wc -w /doc/diccionario.txt
```

```
ls -l | wc -l // Cuenta las líneas del directorio donde este situado.
```

# BÚSQUEDAS DE FICHEROS Y PATRONES

## Comando grep

(Globally Regular Expressions Pattern) Busca patrones en ficheros. Por defecto devuelve todas las líneas que contienen un patrón determinado en uno o varios ficheros.

Sintaxis: `grep [opciones] <patrón> [ficheros]`

### Algunas opciones

- c Devuelve sólo la cantidad de líneas que contienen al patrón.
  - H Imprime además de las líneas, el nombre del fichero donde se encontró el patrón.
  - i Ignora las diferencias entre mayúsculas y minúsculas.
  - l Cuando son múltiples ficheros sólo muestra los nombres de aquellos donde se encontró al patrón y no las líneas correspondientes.
  - n Imprime el número de cada línea que contiene al patrón.
  - r Busca en un directorio de forma recursiva.
  - v Devuelve las líneas que no contienen el patrón.
- ```
grep linux /usr/share/doc // busca la palabra linux en el directorio doc.
grep root /etc/passwd
grep -n error /var/log/messages
grep -i pepe /etc/passwd
grep -c root /etc/group
ls -lia | grep "carta roja"
grep palabra -r . // busca la palabra de forma recursiva desde el directorio
donde esté situado hasta recorrer todo el sistema.
grep ^hola texto.txt // Busca en el fichero texto.txt el patrón que empiece por
hola.
grep hola$ texto.txt // Busca en el fichero texto.txt el patrón que termine con
hola.
```

## Comando find

Este comando es uno de los más potentes en un sistema Linux. Permite buscar de forma recursiva en un directorio a todos los ficheros que cumplan ciertas condiciones. Las condiciones pueden estar relacionadas con el nombre de los ficheros, el tamaño, los permisos, el tipo, las fechas de acceso y modificación, etc.

Sintaxis: `find [camino] [opciones]`

### Algunas opciones

- name <expresión> Permite especificar patrones para los nombres de los ficheros a buscar.
  - iname <expresión> Permite especificar patrones para los nombres de los ficheros a buscar sin tener en cuenta mayúsculas y minúsculas.
  - type <tipo> Tipo de fichero a buscar. Este puede ser **d** para directorios, **f** para ficheros regulares, **l** para enlaces simbólicos, **b** para dispositivos de bloque, **c** para dispositivos de carácter, **p** para tuberías y **s** para sockets.
- ```
find /etc -name '*.conf' // Busca en /etc todos los
ficheros con extensión .conf
```

```
find . -name "*.txt" -type f -delete  
los archivos con extensión .txt
```

// Borrar repulsivamente todos

## COMPRIMIR Y AGRUPAR FICHEROS

### Comandos gzip y gunzip

Permiten comprimir y descomprimir respectivamente uno o varios ficheros.

Sintaxis:

```
gzip [opciones] <ficheros/directorio>  
gunzip [opciones] <ficheros/directorio>
```

### Algunas opciones

- r Dado un directorio comprime todos los ficheros presentes en él recursivamente.
- 1 a -9 Especifica el grado de la compresión (-1 menor y más rápida -9 mayor y más lenta).
- S Permite especificar un sufijo o extensión para el fichero resultado (por defecto es gz).

```
gzip -9 * // Comprime todos los ficheros del directorio actual (su  
extensión cambia a .gz)  
gunzip big-file.gz // Descomprime el fichero big-file.gz
```

### Comando tar

Este comando por defecto no comprime absolutamente nada, se limita a agrupar varios ficheros en uno solo, sin comprimirlos.

Sintaxis: tar [opciones]<fuentes>

### Algunas opciones

- z Compacta o descompacta el fichero resultante una vez agrupado o desagrupado con gzip y gunzip respectivamente.(Ojo, esta opción la usaremos siempre).
- c Permite crear, agrupar ficheros en uno solo.
- x Permite extraer, desagrupar ficheros.
- v Activa el modo debug, donde se ven todos los mensajes
- f Agrupa o desagrupa en o hacia un fichero y no utilizando la salida o entrada estándar como es por defecto.
- l Lista el contenido de un fichero agrupado.

```
// Crea un fichero grande donde estarán agrupados todos los ficheros del directorio  
actual y que además lo comprime.  
tar cvzf grande *  
tar xvzf grande * // Desagrupa en el directorio actual el fichero grande y además  
lo descomprime.  
// Agrupa en el fichero archivo.tar los ficheros passwd, shadow y groups.  
tar cf archivo.tar /etc/passwd /etc/shadow /etc/groups  
tar tf archivo.tar // Muestra el contenido de archivo.tar  
tar xf archivo.tar // Desagrupa el fichero archivo.tar  
tar -tf archivo.tar // Lista el contenido del archivo.tar
```

### Comandos zip y unzip

Comprime y descomprime un fichero o directorio.

Sintaxis:

```
zip [archivo.zip] [ficheros_a_comprimir]
```

```
unzip [archivo.zip]
```

```
zip -r ejemplo.zip ejemplo // Comprime todo el contenido de ejemplo de forma recursiva dentro del archivo ejemplo.zip
```

```
unzip -v archivo.zip // Muestra el contenido de archivo.zip
```

## REDES

### Comando ping

Permite enviar paquetes ICMP (Internet Control Message Protocol) a otra computadora, con el objetivo de saber si esta es alcanzable a través de la red.

Este comando se ejecuta por defecto sostenidamente por lo que **para interrumpirlo se debe pulsar Ctrl-c**.

Sintaxis: ping [opciones] <máquina>

#### Algunas opciones

-c <n> Envía n paquetes exactamente.

-i <n> Espera n segundos entre los envíos.

-s <n> Envía paquetes de n bytes. Se le suman los 8 bytes del header del paquete ICMP.

-q Sólo despliega el sumario final.

```
ping www.google.es // Envía un ping contra el dominio de google.es
```

```
ping 192.168.1.25 // Envía un ping contra una dirección IP.
```

### Comando ifconfig

Permite configurar por parte del usuario root las interfaces de red. Los usuarios distintos de root lo pueden invocar también con fines informativos.

Sin argumentos ifconfig despliega información acerca de la configuración y funcionamiento actuales de las interfaces de red activas.

Sintaxis: ifconfig [opciones]

#### Algunas opciones

-s Muestra un resumen de los interfaces.

```
ifconfig eth0 // Muestra información solo sobre la interfaz eth0.
```

```
ifconfig eth0 up // Levanta la tarjeta de red.
```

```
ifconfig eth0 down // Tira la tarjeta de red.
```

**ifconfig** puede usarse también para configurar los interfaces de red:

```
ifconfig eth0 hw ether 00:30:CA:52:0A:F0 // Cambia la dirección MAC de la interfaz.
```

```
ifconfig eth0 address 192.168.100.2 // Cambia la ip.
```

```
ifconfig wlan0 netmask 255.255.255.0 // Cambia la mascara de red.
```



### Comando hostname

Envía paquetes ICMP (Internet Control Message Protocol) a otro host, con el objetivo de saber si este es alcanzable a través de la red.

Este comando se ejecuta por defecto sostenidamente por lo que para interrumpirlo se debe hacer Ctrl-c

Sintaxis: `hostname [nombre_equipo]`

```
hostname          // Imprime el nombre del equipo
hostname masquweb // Nombra el equipo como masquweb.
```

El nombre del equipo se almacena en el fichero `/etc/host.conf`

### Comando traceroute

Traza la ruta que siguen los paquetes desde el origen hasta el destino, indicando los saltos que existen entre los dos puntos. También existe el comando **tracpath**, para ejecutar este comando no son necesarios privilegios de root.

Sintaxis: `traceroute [opciones] hostname/ip`

```
traceroute www.miweb.es
```

### Comando who

Muestra los usuarios conectados al sistema ya sea local o remotamente.

Sintaxis: `who [opciones] [fichero] [am I]`

desconocido). Si lleva menos de un minuto pone un punto y si es más de 24 horas la cadena 'old'.

-i Imprime además para cada usuario conectado que tiempo lleva sin interactuar con el sistema (idle time).

#### Algunas opciones

-H Imprime un encabezamiento para las columnas.

-w Indica si está activada o no la posibilidad de recibir mensajes por parte de cada usuario conectado (+ indica que si, - indica que no y ?

-q Sólo muestra los logins de los usuarios conectados y la cantidad total de ellos.

```
who
```

```
who am I
```

### Comando netstat

Muestra información sobre las conexiones de red, las tablas de enrutamiento, las estadísticas de los interfaces de red, etc.

Sintaxis: `netstat [opciones]`

#### Algunas opciones

-i Imprime un encabezado para las columnas.

-r Mostrar la tabla de enrutamiento (similar al comando route).

-a Para visualizar todas las conexiones activas en el sistema, tanto TCP como UDP.

```

-l Muestra los sockets a la escucha de servidores.
-t Mostrar solo las conexiones TCP activas.
-u Mostrar solo las conexiones UDP activas.
// Aplicaciones instaladas y "escuchando" en la red usando un puerto tcp.
netstat -l -t
// Aplicaciones instaladas y "escuchando" en la red usando un puerto udp.
netstat -l -u
// Muestra todas las conexiones activas de forma paginada. (pulsar q para terminar)
netstat -a | less

```

### Comando wget

Permite descargar ficheros de la red (incluida Internet) de forma no interactiva, mediante los protocolos https, ftp, ssh, etc.

Sintaxis: `wget [opciones] [url]`

#### Algunas opciones

```

-r Descarga recursivamente, buscando en los subdirectorios del sitio.
-c Continúa una descarga que se interrumpió anteriormente.
-o Registra mensajes en un fichero (logs).
--tries=n Establece el número de reintentos.
--limit-rate=n Establece una velocidad máxima de descargas en KB.
-m, --mirror Hace una copia completa de el directorio o url de el sitio que indiquemos.
-P Sirve para indicar el sitio donde descargar los archivos.
wget --trues=10 https://www.miweb.es/images/cartel.jpg
// Descargar página completa en el directorio Descarga y crear un registro en el
archivo log.txt
wget -P/descarga https://www.miweb.com -o log.txt
// Nos autentica en el servidor ftp y descarga el documento.
wget -c --user=usuario --password=contraseña ftp://servidor.es/documento.odt
// Crear un backup de nuestra pagina
wget --mirror /descarga https://www.miweb.com

```

### Comando nslookup

Permite saber la ip de un host a través de su nombre y viceversa, realizando consultas al servidor de nombres DNS.

Sintaxis: `nslookup [url/ip]`

```
nslookup www.miweb.es
```

### Comando ssh

Permite acceder a otra maquina a través de la red, ejecutar comandos y mover ficheros entre dos máquinas. Este comando ofrece conexión cifrada entre dos sistemas proporcionando una conexión segura.

Sintaxis: `ssh [opciones] <usuario@maquinaremota>`

### Algunas opciones

-p Permite especificar el puerto ssh  
-l Permite especificar el nombre de usuario (login name).  
// Si la conexión es correcta pedirá usuario y contraseña.  
ssh -p 22 jose@midominio.es  
// Conectarse al servidor usando el puerto 22  
ssh -p 22 jose@servidor.es  
// Conexión del usuario jose a la maquina remota usando el puerto 2220 y la dirección ip  
ssh -l jose -p 2220 192.168.20.31

#### Comando scp

Permite copiar ficheros entre dos máquinas. Utiliza ssh para la transmisión de datos.

Sintaxis: scp [opciones] <usuari@maquinaremota>

### Algunas opciones

-r Permite copiar directorios recursivamente.  
-P Permite especificar el puerto scp  
// Copiar todo el contenido del directorio local a un directorio remoto.  
scp -r /origen/\* usuario@dominio:/ruta/destino/  
// Copiar el archivo.txt de una maquina remota a nuestra maquina local.  
scp usuario@maquinaremota:/ruta/archivo.txt /ruta/destino  
// Copiar multiples archivos en la maquina remota limitando el ancho de banda a 50Kbit/s.  
scp -r -P 2222 /origen/\*.php usuario@dominio:/ruta/destino/

#### Comando ftp

Permite la conexión con un servidor FTP.

Sintaxis:

ftp <servidor|ip> [puerto]  
ftp <usuario@servidor|ip> [puerto]

ftp 192.168.20.7  
ftp usuario@maquinaremota 2100 // Conexión ftp usando el puerto 2100

## DESCONEXIÓN DEL SISTEMA

#### Comando exit

Permite terminar el shell actual. Si se tiene un único shell es equivalente a desconectarse del sistema, pero si se está en un subshell sólo se terminará este, retornando al shell anterior.

Sintaxis: exit

#### Comando logout

Permite desconectarse del sistema a partir de un login shell.

Sintaxis: logout

## **OTROS**

### **Comando clear**

Limpia el terminal, y coloca el prompt al principio.

Sintaxis: `c`lear