

Repasso de Linux

Repasso de Linux

Capítulo 2 - Ficheros y directorios

Comandos `pwd`, `ls`, `cd`, `mkdir`

Comando `pwd`

Comando `ls`

Comando `ls -a`

Comando `ls -l`

Comando `ls -h`

Comando `cd`

Comando `mkdir`

Visualización de ficheros (`cat`, `more`, `less`, `head`, `tail`)

Comando `cat`

Comando `more`

Comando `less`

Comandos `head` y `tail`

EDICIÓN DE FICHEROS (touch, vi, ee, mcedit)

Ejercicios del capítulo 2

CAPÍTULO 3

Caracteres comodín

Copia y borrado de ficheros (`cp`, `mv`, `rm`)

Copia y borrado de directorios(`cp`, `mv`, `rm`)

Ejercicios Capítulo 3

Capítulo 4, Grupos, usuarios y permisos

Permisos

¿Quienes somos?(`whoami`, `groups`)

Gestión de grupos (`groupadd`, `groupdel`, `groupmod`)

Gestión de usuarios (`adduser`, `userdel`, `usermod`)

Cambio de grupo y de dueño (`chown`, `chgrp`)

Cambio de privilegios (`chmod`)

Resumen capítulo 4

Comandos de Red en Linux

1. `ip a`
2. `ifconfig`
3. `ping`
4. `nslookup`
5. `netstat`
6. `curl`
7. `hostname`
8. `whois`
9. `ip route`

Ejemplos de uso

CUESTIONES

Capítulo 2 - Ficheros y directorios

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.

Comandos `pwd`, `ls`, `cd`, `mkdir`

Comando `pwd`

Muestra cuál es el directorio de trabajo actual

```
$ pwd
```

```
david@cliente24:~/Escritorio$ pwd  
/home/david/Escritorio  
david@cliente24:~/Escritorio$
```

Comando `ls`

Muestra el contenido del directorio actual. Por defecto los archivos ocultos no se muestran.

```
$ ls
```

```
david@cliente24:~/Escritorio$ ls  
'Nueva carpeta'  
david@cliente24:~/Escritorio$
```

Comando `ls -a`

Comando que muestra todos los archivos, incluyendo los ocultos.

```
$ls -a
```

```
david@cliente24:~/Escritorio$ ls -a  
. .. 'Nueva carpeta'
```

Comando `ls -l`

Muestra un listado detallado, con la última fecha de modificación de cada archivo, el tamaño, etc.

```
$ls -l
```

```
david@cliente24:~/Escritorio$ ls -l
total 4
drwxrwxr-x 2 david david 4096 sep 25 16:51 'Nueva carpeta'
```

Comando `ls -h`

```
$ls -h
```

```
david@cliente24:~/Escritorio$ ls -h
'Nueva carpeta'    txt
```

Comando `cd`

Permite cambiar de directorio.

```
$ cd
```

```
david@cliente24:~/Escritorio$ cd
```

Comando `mkdir`

Comando usado para crear directorios.

```
$ mkdir nombreCarpeta
```

```
david@cliente24:~$ mkdir carpetaNueva
david@cliente24:~$ cd carpetaNueva/
david@cliente24:~/carpetaNueva$
```

Visualización de ficheros (`cat`, `more`, `less`, `head`, `tail`)

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.

Comando `cat`

Comando que muestra por pantalla el contenido de un fichero y, cuando termina, el usuario está otra vez de vuelta en la línea de comandos.

```
$ cat  
$cat /var/log/dmesg
```

`cat /var/log/dmesg` muestra el contenido del fichero dmesg que está dentro del directorio `/var/log`.

```
david@cliente24:~/carpetaNueva$ cat  
cat/var/log/dmesg  
cat/var/log/dmesg
```

Comando `more`

Hace lo mismo que `cat`, a diferencia de que muestra el fichero pantalla a pantalla. Llena de texto la pantalla y espera a que el usuario pulse la tecla `Space` para pasar a la siguiente.

```
$ more  
$ more /var/log/dmesg
```

```
david@lct-test:~$ more /var/log/dmesg
[ 0.000000] kernel: Linux version 6.14.0-32-generic (buildd@lcy02-amd64-047)
(x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0, GNU Binu
tts for Ubuntu) 2.42) #32~24.04.1-Ubuntu SMP PREEMPT_DYNAMIC Tue Sep 2 14:21:0
4 UTC 2 (Ubuntu 6.14.0-32.32-24.04.1-generic 6.14.11)
[ 0.000000] kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-6.14.0-32-generic
root=UUID=c822b659-83b9-46e6-bc5c-939d7e56704f ro quiet splash
[ 0.000000] kernel: KERNEL supported cpus:
[ 0.000000] kernel: Intel GenuineIntel
[ 0.000000] kernel: AMD AuthenticAMD
[ 0.000000] kernel: Hygon HygonGenuine
[ 0.000000] kernel: Centaur CentaurHauls
[ 0.000000] kernel: zhaoxin Shanghai
[ 0.000000] kernel: BIOS-provided physical RAM map:
[ 0.000000] kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] us
able
[ 0.000000] kernel: BIOS-e820: [mem 0x0000000000000009fc00-0x000000000009ffff] re
served
[ 0.000000] kernel: BIOS-e820: [mem 0x000000000000f0000-0x0000000000ffff] re
served
[ 0.000000] kernel: BIOS-e820: [mem 0x00000000000100000-0x00000000dfffffff] us
able
[ 0.000000] kernel: BIOS-e820: [mem 0x000000000dfff0000-0x00000000dfffffff] AC
PI data
[ 0.000000] kernel: BIOS-e820: [mem 0x000000000fc00000-0x000000000fc0fff] reserved
[ 0.000000] kernel: BIOS-e820: [mem 0x000000000fe00000-0x000000000fe0fff] reserved
[ 0.000000] kernel: BIOS-e820: [mem 0x000000000ffc0000-0x000000000ffcffff] reserved
[ 0.000000] kernel: BIOS-e820: [mem 0x00000000010000000-0x000000001988ffff] us
able
[ 0.000000] kernel: NX (Execute Disable) protection: active
[ 0.000000] kernel: APIC: Static calls initialized
[ 0.000000] kernel: SMBIOS 2.5 present.
[ 0.000000] kernel: DMI: innotek Gmbh VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[ 0.000000] kernel: DMI: Memory slots populated: 0/0
[ 0.000000] kernel: Hypervisor detected: KVM
[ 0.000000] kernel: kvm-clock: Using msrs 4b564d01 and 4b564d00
[ 0.000001] kernel: kvm-clock: using sched offset of 4892543437 cycles
[ 0.000003] kernel: clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles: 0x1cd42e4dff, max_idle_ns: 881590591483 ns
[ 0.000006] kernel: tsc: Detected 2688.098 MHz processor
[ 0.000066] kernel: e828: update [mem 0x000000000-0x0000ffff] usable ==> reserved
[ 0.000069] kernel: e828: remove [mem 0x000a000-0x000ffff] usable
[ 0.000072] kernel: last_pfn = 0x1099000 max_arch_pfn = 0x400000000
[ 0.000088] kernel: MTRR map: 3 entries (3 fixed + 0 variable; max 19), built from 8 variable MTRRs
[ 0.000090] kernel: x86/PAT: Configuration [0-7]: W0 WC UC UC WB WP UC WT
[ 0.000098] kernel: CPU MTRRs all blank - virtualized system.
[ 0.000099] kernel: last_pfn = 0xe0000 max_arch_pfn = 0x400000000
```

Comando less

Es el más versátil de los tres, ya que permite moverse hacia delante y hacia atrás dentro del fichero.

```
$ less /var/log/dmesg
```

```
[ 0.000000] kernel: Linux version 6.14.0-32-generic (buildd@lcy02-amd64-047)
(x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0, GNU Binu
tts for Ubuntu) 2.42) #32~24.04.1-Ubuntu SMP PREEMPT_DYNAMIC Tue Sep 2 14:21:0
4 UTC 2 (Ubuntu 6.14.0-32.32-24.04.1-generic 6.14.11)
[ 0.000000] kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-6.14.0-32-generic
root=UUID=c822b659-83b9-46e6-bc5c-939d7e56704f ro quiet splash
[ 0.000000] kernel: KERNEL supported cpus:
[ 0.000000] kernel: Intel GenuineIntel
[ 0.000000] kernel: AMD AuthenticAMD
[ 0.000000] kernel: Hygon HygonGenuine
[ 0.000000] kernel: Centaur CentaurHauls
[ 0.000000] kernel: zhaoxin Shanghai
[ 0.000000] kernel: BIOS-provided physical RAM map:
[ 0.000000] kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] us
able
[ 0.000000] kernel: BIOS-e820: [mem 0x0000000000000009fc00-0x000000000009ffff] re
served
[ 0.000000] kernel: BIOS-e820: [mem 0x000000000000f0000-0x0000000000ffff] re
served
[ 0.000000] kernel: BIOS-e820: [mem 0x00000000000100000-0x00000000dfffffff] us
able
[ 0.000000] kernel: BIOS-e820: [mem 0x000000000dfff0000-0x00000000dfffffff] AC
PI data
/var/log/dmesg
```

Comandos head y tail

Permiten mostrar de forma parcial el contenido de un fichero. `Head` muestra la cabezera y `tail` las últimas líneas.

```
$ head  
$ head /boot/grub/menu.lst
```

```
david@cliente24:~/Escritorio$ head /boot/grub/menu.lst
```

```
$ tail  
$ tail /boot/grub/menu.lst
```

```
david@cliente24:~/Escritorio$ tail /boot/GRUB/menu.lst
```

EDICIÓN DE FICHEROS (touch, vi, ee, mcedit)

El comando `touch` permite crear un fichero vacío.

```
$ touch
```

```
david@cliente24:~/Escritorio$ touch  
touch: falta un archivo como argumento
```

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

```
$ee nombreDocumento.txt
```

```
david@cliente24:~/Escritorio$ ee  
ee: no se encontró la orden
```

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 nombreDocumento.txt
```

```
david@cliente24:~/Escritorio$ cat
```

Otro editor simple es `nano`. Si no esta instalado alguno de estos editores, tecleamos `$sudo apt-get install` seguido del nombre del programa.

```
$ sudo apt-get install e
```

```
david@cliente24:~/Escritorio$ cat  
sudo apt-get install ee  
sudo apt-get install ee
```

El programa `mcedit` es una parte de `mc`. Instalaremos `mc`

```
$ sudo apt-get install mc  
$ mcedit nombreDocumento.txt
```

```
sudo apt-get install ee  
sudo apt-get install ee  
sudo apt-get install mc  
sudo apt-get install mc  
mcedit nombreDocumento  
mcedit nombreDocumento
```

Editor de Linux por excelencia: `vi`

```
$ vi nombreDoc.txt
```



Ejercicios del capítulo 2

1. ¿En qué directorio se encuentran los ficheros de configuración del sistema?

En el directorio `/etc`

2. Para entrar en un sistema Linux hace falta a) nombre de usuario, contraseña y dirección IP, b) nombre de usuario y contraseña o c) únicamente una contraseña.

b) nombre de usuario y contraseña

3. Muestra el contenido del directorio actual.

```
$ ls
```

```
david@cliente24:~/Escritorio$ ls  
'Nueva carpeta'
```

4. Muestra el contenido del directorio que está justo a un nivel superior.

```
$ ls ..
```

```
david@cliente24:~/Escritorio$ ls ..  
carpetaNueva Documentos Imágenes Plantillas snap  
Descargas Escritorio Música Público Vídeos
```

5. ¿En qué día de la semana naciste?, utiliza la instrucción `cal` para averiguarlo.

```
$ cal julio 2002
```

```
david@cliente24:~/Escritorio$ cal julio 2002
                Julio 2002
do lu ma mi ju vi sa
      1  2  3  4  5  6
      7  8  9 10 11 12 13
     14 15 16 17 18 19 20
     21 22 23 24 25 26 27
     28 29 30 31
```

6. Muestra los archivos del directorio [/bin](#)

```
ls /bin
```

```
david@cliente24:~/Escritorio$ ls /bin
['
aa-enabled
aa-exec
aa-features-abi
aconnect
acpidbg
add-apt-repository
addpart
addr2line
airscan-discover
alsabat
alsaloop
alsamixer
alsatplg
alsaucm
amidi
amixer
apg
apgbfm
aplay
aplaymidi
aport-bug
aport-cli
aport-collect
aport-unpack
appres
appstreamcli
apropos
apt
apt-add-repository
ncal
nc.openbsd
negn
netaddr
netcat
networkctl
networkd-dispatcher
newgrp
gettext
nice
nisdomainname
nl
nm
nm-applet
nmcli
nm-connection-editor
nm-online
nmtui
nmtui-connect
nmtui-edit
nmtui-hostname
nohup
notify-send
nproc
nroff
nsenter
nslookup
nstat
nsupdate
ntfs-3g
```

7. Suponiendo que te encuentras en tu directorio personal ([/home/nombre](#)), muestra un listado del contenido de [/usr/bin](#)

- a) con una sola línea de comando, b) moviéndote paso a paso por los directorios y c) con dos líneas de comandos.

```
david@cliente24:~/Escritorio$ ls /bin
['[',
 aa-enabled          ncal
 aa-exec             nc.openbsd
 aa-features-abi    neqn
 aconnect            netaddr
 acpidbg             netcat
 add-apt-repository networkd-dispatcher
 addpart              newgrp
 addr2line            ngettext
 airscan-discover    nice
 alsabat              nisdomainname
 alsaloop             nl
 alsamixer            nm
 alsatplg             nm-applet
 alsaucm              nmcli
 amidi                nm-connection-editor
 amixer               nm-online
 apg                 nmtui
 apgbfm               nmtui-connect
 aplay               nmtui-edit
 aplaymidi            nmtui-hostname
 apport-bug           nohup
 apport-cli            notify-send
 apport-collect        nproc
 apport-unpack         nroff
 appres               nsenter
 appstreamcli          nslookup
 apropos              nstat
 apt                 nsupdate
 apt-add-repository  ntfs-3g
```

a)

```
ls /bin
```

```
dc
david@cliente24:~/Escritorio$ cd ..
david@cliente24:~$ cd ..
david@cliente24:/home$ cd usr
bash: cd: usr: No existe el archivo o el directorio
b) david@cliente24:/home$ cd bin
bash: cd: bin: No existe el archivo o el directorio
david@cliente24:/home$ ls
david
* * * * *
```

```
$ cd ..
cd usr
cd bin
```

c)

```
david@cliente24:~/Escritorio$ ls /usr/bin
['
aa-enabled
aa-exec
aa-features-abi
aconnect
acpidbg
add-apt-repository
addpart
addr2line
ncal
nc.openbsd
neqn
netaddr
netcat
networkctl
networkd-dispatcher
newgrp
gettext
```

```
ls /usr/bin
```

8. Muestra todos los archivos que hay en `/etc` y todos los que hay dentro de cada subdirectorio, de forma recursiva (con un solo comando).

```
david@cliente24:~/Escritorio$ ls -R /etc
/etc:
adduser.conf          host.conf           profile
alsa                  hostname            profile.d
alternatives          hosts               protocols
anacrontab            hosts.allow         prueba.txt
apg.conf              hosts.deny          pulse
apm                  ifplugd             python3
apparmor              init                python3.12
apparmor.d            init.d              rc0.d
apport                init.d              rc1.d
```

```
$ ls -R /etc
```

9. Muestra todos los archivos del directorio `/usr/X11R6/bin` ordenados por tamaño (de mayor a menor). Sólo debe aparecer el nombre de cada fichero, sin ninguna otra información adicional.

```
david@cliente24:~/Escritorio$ ls -S /usr/X11R6/bin
```

```
$ ls -S /usr/X11R6/bin
```

10. Muestra todos los archivos del directorio `/etc` ordenados por tamaño (de mayor a menor) junto con el resto de características, es decir, permisos, tamaño, fechas de la última modificación, etc. El tamaño de cada fichero debe aparecer en un formato “legible”, o sea, expresado en Kb, Mb, etc.

```
david@cliente24:~/Escritorio$ ls -R /etc
/etc:
adduser.conf          host.conf           profile
alsa                  hostname            profile.d
alternatives          hosts               protocols
anacrontab            hosts.allow        prueba.txt
apg.conf              hosts.deny         pulse
apm                  ifplugd             python3
apparmor              init                rc0.d
apparmor.d            init.d              rc1.d
apport                initramfs-tools   rc2.d
apt                  inputrc             rc3.d
avahi
```

```
$ ls -Slh /etc
```

11. Muestra todos los archivos del directorio `/bin` ordenados por tamaño (de menor a mayor). Sólo debe aparecer el tamaño y el nombre de cada fichero, sin ninguna otra información adicional. El tamaño de cada fichero debe aparecer en un formato “legible”, o sea, expresado en Kb, Mb, etc.

```
david@cliente24:~/Escritorio$ ls -Sshr /bin
total 225M
    0  X11
    0  xzcat
    0  unxz
    0  ghostscript
    0  sleep
```

```
$ ls -Sshr /bin
```

12. Muestra el contenido del directorio raíz utilizando como argumento de `ls` una ruta absoluta.

```
david@cliente24:~/Escritorio$ ls /  
bin          home          mnt      sbin usr-is-merged  usr  
bin usr-is-merged lib          opt      snap           var  
boot         lib64         proc     srv  
cdrom        lib usr-is-merged root    swap.img  
dev          lost+found    run     sys  
etc          media         sbin    tmp
```

```
$ ls /
```

13. Muestra el contenido del directorio raíz utilizando como argumento de `ls` una ruta relativa.
Suponemos que el directorio actual es `/home/elenas/documentos`.

```
david@cliente24:~/Escritorio$ ls ../../..  
bin          home          mnt      sbin usr-is-merged  usr  
bin usr-is-merged lib          opt      snap           var  
boot         lib64         proc     srv  
cdrom        lib usr-is-merged root    swap.img  
dev          lost+found    run     sys  
etc          media         sbin    tmp
```

```
$ ls ../../..
```

14. Crea el directorio `gastos` dentro del directorio personal

```
david@cliente24:~/Escritorio$ mkdir gastos  
david@cliente24:~/Escritorio$
```

```
$ mkdir gastos
```

15. ¿Qué sucede si se intenta crear un directorio dentro de `/etc`

Solo podria hacerlo un usuario root.

16. Muestra el contenido del fichero `/etc/fstab`

```
david@cliente24:~/Escritorio$ cat /etc/bash.bashrc
# System-wide .bashrc file for interactive bash(1) shells.

# To enable the settings / commands in this file for login shells as well,
# this file has to be sourced in /etc/profile.

# If not running interactively, don't do anything
[ -z "$PS1" ] && return

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize
```

```
$ cat /etc/bash.bashrc
```

17. Muestra las 10 primeras líneas del fichero `/etc/bash.bashrc`

```
david@cliente24:~/Escritorio$ head /etc/bash.bashrc
# System-wide .bashrc file for interactive bash(1) shells.

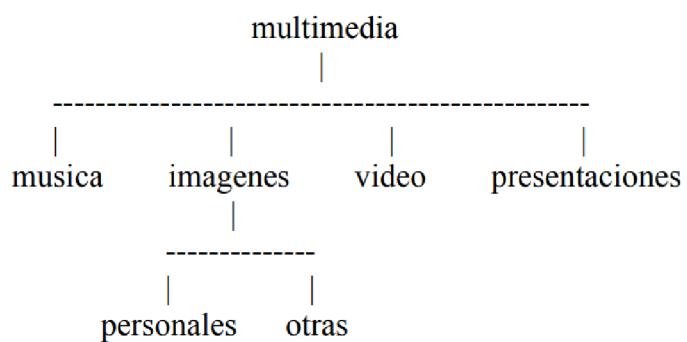
# To enable the settings / commands in this file for login shells as well,
# this file has to be sourced in /etc/profile.

# If not running interactively, don't do anything
[ -z "$PS1" ] && return

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
```

18. `$ head /etc/bash.bashrc`

19. Crea la siguiente estructura de directorios dentro del directorio de trabajo personal:



```

david@cliente24:~/Escritorio$ mkdir multimedia
david@cliente24:~/Escritorio$ cd multimedia/
david@cliente24:~/Escritorio/multimedia$ multimedia$ mkdir musica imagenes video
    presentaciones
multimedia$: no se encontró la orden
david@cliente24:~/Escritorio/multimedia$ multimedia$ cd imagenes/
multimedia$: no se encontró la orden
david@cliente24:~/Escritorio/multimedia$ /multimedia/imagenes$ mkdir personales
otras
bash: /multimedia/imagenes$: No existe el archivo o el directorio
  
```

```

~$ mkdir multimedia
~$ cd multimedia/
~/multimedia$ mkdir musica imagenes video presentaciones
~/multimedia$ cd imagenes/
~/multimedia/imagenes$ mkdir personales otras
  
```

19. Crea un fichero vacío dentro del directorio `musica`, con nombre `estilos_favoritos.txt`

```

david@cliente24:~/Escritorio$ cd
david@cliente24:~$ cd multimedia/musica
bash: cd: multimedia/musica: No existe el archivo o el directorio
david@cliente24:~$ touch estilos_favoritos.txt
david@cliente24:~$ 
  
```

```

$ cd
$ cd multimedia/musica
$ touch estilos_favoritos.txt
  
```

20. Utiliza tu editor preferido para abrir el fichero `estilos_favoritos.txt` e introduce los estilos de música que más te gusten. Guarda los cambios y sal.

```
$ vi favoritos.txt  
$ i  
$ techno  
$ groove  
$ hardbounce  
$ ESC:w  
$ :q
```

21. Muestra todo el contenido de `estilos_favoritos.txt`

```
$ cat estilos_favoritos.txt
```

22. Muestra las 3 primeras líneas de `estilos_favoritos.txt`

```
$ head -n3 estilos.favoritos.txt
```

23. Muestra la última línea de `estilos_favoritos.txt`

```
$ tail -n1 estilos_favoritos.txt
```

24. Muestra todo el contenido del fichero `estilos_favoritos.txt` excepto la primera línea. Se supone que no sabemos de antemano el número de líneas del fichero.

```
$ tail -n+2 estilos_favoritos.txt
```

CAPÍTULO 3

Caracteres comodín

Muchas veces necesitamos realizar acciones sobre muchos archivos o directorios al mismo tiempo:

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

```
david@cliente24:~/Escritorio$ cat docu1 docu2 docu3 docu4 docu5 docu6
cat: docu1: No existe el archivo o el directorio
cat: docu2: No existe el archivo o el directorio
cat: docu3: No existe el archivo o el directorio
cat: docu4: No existe el archivo o el directorio
cat: docu5: No existe el archivo o el directorio
cat: docu6: No existe el archivo o el directorio
```

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

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

```
david@cliente24:~/Escritorio$ cat fich[1-6]
cat: 'fich[1-6]': No existe el archivo o el directorio
david@cliente24:~/Escritorio$ █
```

Para mostrar solo el contenido de todos los ficheros que empiezan por fich se hace:

```
$ cat fich*
```

```
david@cliente24:~/Escritorio$ cat fich*
cat: 'fich*': No existe el archivo o el directorio
david@cliente24:~/Escritorio$ █
```

Para mostrar todos los ficheros que empiezan por la letra a y terminan por las letras dentro del directorio </usr/bin>

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

```
david@cliente24:~/Escritorio$ ls /usr/bin/a*
/usr/bin/appres   /usr/bin/apt-extracttemplates  /usr/bin/as
/usr/bin/apropos  /usr/bin/apt-sortpkgs
```

? representa un carácter cualquiera. La siguiente sentencia muestra todos los ficheros del directorio /usr/bin cuyo nombre empieza por g , sigue cualquier carácter, a continuación sigue una o y acaba con cualquier cadena de caracteres.

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

```
david@cliente24:~/Escritorio$ ls /usr/bin/g?o*
/usr/bin/gcore          /usr/bin/gnome-logs
/usr/bin/gcov           /usr/bin/gnome-power-statistics
/usr/bin/gcov-13         /usr/bin/gnome-session
/usr/bin/gcov-dump       /usr/bin/gnome-session-inhibit
/usr/bin/gcov-dump-13    /usr/bin/gnome-session-properties
/usr/bin/gcov-tool        /usr/bin/gnome-session-quit
/usr/bin/gcov-tool-13    /usr/bin/gnome-shell
```

Copia y borrado de ficheros (cp, mv, rm)

El comando cp sirve para copiar ficheros. Se puede copiar uno o muchos.

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

```
david@cliente24:~/Escritorio$ cp /etc/hosts /home/alumno/pruebas/
cp: no se puede crear el fichero regular '/home/alumno/pruebas/': No existe el archivo o el directorio
```

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

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

```
david@cliente24:~/Escritorio$ cp *.opt textos/
cp: no se puede efectuar 'stat' sobre '*.opt': No existe el archivo o el directorio
```

Cuando se quiere especificar como directorio destino el directorio actual se usa el carácter .

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

```
david@cliente24:~/Escritorio$ cp /usr/bin/g*  
cp: target '/usr/bin/gzip': No es un directorio
```

El comando `mv` sirve para mover y para cambiar de nombre.

```
$ 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:

```
$ cd Documentos/  
/ Documentos$ mkdir correspondencia  
/Documentos$ mv carta.txt correspondencia/carta01.txt
```

El comando `rm` se usa para borrar ficheros. Es importante destacar que estos ficheros no se envían a una papelera así que **NO SE PUEDEN RECUPERAR AL BORRARSE**

```
$ rm *.txt
```

Copia y borrado de directorios(cp, mv, rm)

Un directorio puede contener muchos ficheros y, otros directorios con mas ficheros y directorios. Para copiar un fichero completo hay que indicarlo con la opcion `-R`. A esto se le suele llamar **copiar de forma recursiva**

```
$ mkdir multimedia2
$ cp multimedia/* multimedia2
cp: se omite el directorio «multimedia/imagenes»
cp: se omite el directorio «multimedia/musica»
cp: se omite el directorio «multimedia/presentaciones»
cp: se omite el directorio «multimedia/video»
~$ ls multimedia2
```

Se hizo una copia del contenido del directorio `multimedia` al `multimedia2` pero sin añadir todo lo que viene dentro porque no se hizo la *copia recursiva*.

Ahora vamos a hacerlo 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:

multimedia/video:
```

Hemos copiado tanto la estructura de directorios como los contenidos de cada uno de ellos.

El comando `mv` funciona de forma parecida a `cp`, pero mueve en vez de copiar. Para renombrar funciona igual que con los ficheros.

```
$ mv multimedia2 multimedia_copia
```

Cambia el nombre de `multimedia2` a `multimedia_copia`

Con `rm` se pueden borrar directorios pero tiene que ser de forma recursiva

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

Hemos añadido la opción `-f` que hace que no se nos pida confirmación por cada elemento que se quiere borrar.

Ejercicios Capítulo 3

1. Muestra todos los archivos del directorio actual que son imágenes jpg.

```
$ ls *.jpg
```

- 2.. Muestra todos los archivos del directorio `/usr/bin` que empiecen por la letra j.

```
$ ls /usr/bin/j*
```

3. Muestra los archivos que empiecen por k y tengan una a en la tercera posición, dentro del directorio `/usr/bin`.

```
$ls /usr/bin/k?a*
```

4. Muestra los archivos del directorio `/bin` que terminen en n.

```
$ls /bin/*n
```

5. Muestra todos los archivos que hay en `/etc` y todos los que hay dentro de cada subdirectorio, de forma recursiva.

```
$ls -R /etc
```

6. Crea un directorio en tu directorio de trabajo con nombre prueba. Copia el archivo gzip del directorio /bin al directorio prueba. Crea un duplicado de gzip con nombre gzip2 dentro de prueba.

```
~$ mkdir prueba  
~$ cp /bin/gzip prueba  
~$ cd prueba/  
~/prueba$ cp gzip gzip2
```

7. Cambia el nombre de prueba a prueba2. Crea prueba3 en el mismo nivel que prueba2 y mueve todos los ficheros de prueba2 a prueba3. Borra prueba2.

```
~$prueba$ cd ..  
~$ mv prueba prueba2  
~$ mkdir prueba3  
~$ mv prueba2/* prueba3/  
~$ rmdir prueba2
```

8. Crea un fichero vacío con nombre “?Hola caracola?”. ¿Se puede? En caso de que se pudiera, ¿sería recomendable poner nombres así? Razona la respuesta.

Si usando \ para indicar caracteres especiales

No se debería de usar caracteres raros.

9. Crea un directorio con nombre multimedia_pruebas y copia en él todo el contenido del directorio multimedia. A continuación crea en multimedia/video/ dos ficheros, uno con nombre peliculas.txt y otro con nombre actores.txt. Edita el fichero peliculas.txt e introduce el nombre de tu película favorita. A continuación, crea en multimedia_pruebas/video/ otro fichero que también tenga por nombre peliculas.txt, editalo y esta vez escribe el nombre de tus cinco películas favoritas. Ahora haz una copia de todo el contenido de multimedia en multimedia_prueba de tal forma que sólo se copien los contenidos nuevos, es decir, si hay coincidencia en el nombre de un archivo se respetará el que se haya modificado más recientemente. Para comprobar que se ha hecho todo correctamente, basta mirar si en multimedia_prueba/video está el archivo vacío actores.txt y además el archivo peliculas.txt debe contener 5 películas y no 1.

```
~$ mkdir multimedia_pruebas  
~$ cp -R multimedia/* multimedia_pruebas/  
~$ cd multimedia /video/  
~/multimedia/video$ touch peliculas.txt actores.txt  
~/multimedia/video$ vi peliculas.txt  
~/multimedia/video$ cd  
~$ cd multimedia_pruebas/video/  
~/multimedia_pruebas/video$ vi peliculas.txt  
~/multimedia_pruebas/video$ cd  
~$ cp -Ru multimedia/* multimedia_pruebas/
```

10. Borra el directorio multimedia/imagenes/otras. El sistema debe pedir al usuario que confirme el borrado.

```
$ rm -Ri multimedia/imagenes/otras/
```

11. Mueve el archivo peliculas.txt, que está dentro de multimedia/video, al directorio que está justo a un nivel superior. Ahora el archivo debe llamarse mis_peliculas.txt en lugar de películas.

```
~$ cd multimedia/video/  
~/multimedia/video$ mv peliculas.txt ../mis_peliculas.txt
```

Capítulo 4, Grupos, usuarios y permisos

El **SUPERUSUARIO**, administrador del sistema o el **root** es un usuario que tiene privilegios para cambiar la configuración, borrar y crear ficheros en cualquier directorio, crear nuevos grupos y usuarios, etc. Hay que ser responsable al trabajar como **SUPERUSUARIO**

Hacer un **root** :

```
$ touch /etc/prueba.txt
touch: no se puede efectuar `touch' sobre «/etc/prueba.txt»: Permiso denegado
$ sudo touch /etc/prueba.txt
$ ls /etc/prueb*
/etc/prueba.txt
```

```
david@cliente24:~/Escritorio$ touch /etc/prueba.txt
touch: no se puede efectuar `touch' sobre '/etc/prueba.txt': Permiso denegado
david@cliente24:~/Escritorio$ sudo touch /etc/prueba.txt
[sudo] contraseña para david:
david@cliente24:~/Escritorio$ ls /etc/prueb*
/etc/prueba.txt
```

Lo de permiso denegado es porque un usuario sin privilegios no puede hacer eso.

Permisos

Para conocer información sobre grupos, usuarios y permisos, usamos el comando `ls` junto con la opción `-l`. Veamos los permisos del fichero `whatis` que está en el directorio `/usr/bin`

```
$ ls -l /usr/bin/whatis
-rwxr-xr-x 1 root root 87792 2008-03-12 14:24 /usr/bin/whatis
```

```
david@cliente24:~/Escritorio$ ls -l /usr/bin/whatis
-rwxr-xr-x 1 root root 48416 abr  8 2024 /usr/bin/whatis
```

En la primera columna están los **permisos**, en la tercera el **usuario** y en la cuarta el nombre del **grupo**

-	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 .

Tipo de fichero

<i>Tipo de fichero</i>	
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.

¿Quienes somos?(whoami, groups)

Podemos usar **su** para ejecutar comandos como otro user distinto, siempre que sepamos la contraseña.

```
$ whoami
luisjose
$ su alumno
Contraseña:
$ whoami
alumno
```

```
david@cliente24:~/Escritorio$ whoami
david
david@cliente24:~/Escritorio$ su alumno:
su: user alumno: does not exist or the user entry does not contain all the required fields
```

Para volver a ser el user original, solo hay que pulsar **exit**

```
$ whoami  
alumno  
$ exit  
exit  
$ whoami  
luisjose
```

Con el comando `groups` se puede ver a qué grupo pertenemos.

```
luisjose@luisjose-xps1330:~$ groups  
luisjose adm dialout cdrom floppy audio dip video plugdev scanner lpadmin admin  
netdev powerdev sambashare
```

Se pueden especificar uno o mas usuarios detras de `groups`. Eso nos dirá a que grupo pertenece cada uno de ellos.

```
luisjose@luisjose-xps1330:~$ groups alumno root  
alumno : alumno  
root : root
```

Gestión de grupos (`groupadd`, `groupdel`, `groupmod`)

Esos comandos permiten crear, borrar y modificar grupos respectivamente.

Crearemos los grupos `oficina_malaga`, `oficina_jaen` y `oficina_madrid`

```
$ groupadd oficina_malaga  
groupadd: incapaz de bloquear el fichero de grupos  
$ sudo groupadd oficina_malaga  
$ sudo groupadd oficina_jaen  
$ sudo groupadd oficina_madrid
```

```
david@cliente24:~/Escritorio$ groupadd oficina_malaga
groupadd: Permission denied.
groupadd: no se pudo bloquear /etc/group, inténtelo de nuevo.
david@cliente24:~/Escritorio$ sudo groupadd oficina_malaga
[sudo] contraseña para david:
david@cliente24:~/Escritorio$ sudo groupadd oficina_jaen
david@cliente24:~/Escritorio$ sudo groupadd oficina_madrid
```

Para corregir un error de escritura, usaremos `groupmod`. Por ejemplo:

```
$ sudo groupmod -n oficina_madrid oficina madrit
```

Para borrar un grupo usaremos `groupdel`

```
$ sudo groupdel oficina_jaen
```

```
david@cliente24:~/Escritorio$ sudo groupdel oficina_jaen
david@cliente24:~/Escritorio$
```

Gestion de usuarios (adduser, userdel, usermod)

La gestión de usuarios, 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:

```
$ sudo bash
```

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`

Daremos de alta dos usuarios en la de malaga, uno en la de madrid y otro que irá y volverá de una a otra.

```
# adduser pedro --ingroup oficina_malaga  
# adduser ana --ingroup oficina_malaga  
  
# adduser berta --ingroup oficina_madrid  
# adduser laura --ingroup oficina_malaga  
# adduser laura oficina_madrid
```

Hemos creado los usuarios y a su vez, los hemos incluido dentro de los grupos correspondientes.

Es importante destacar que hemos `adduser` y no `useradd`. Este último es considerado un comando de bajo nivel y se recomienda usar el primero.

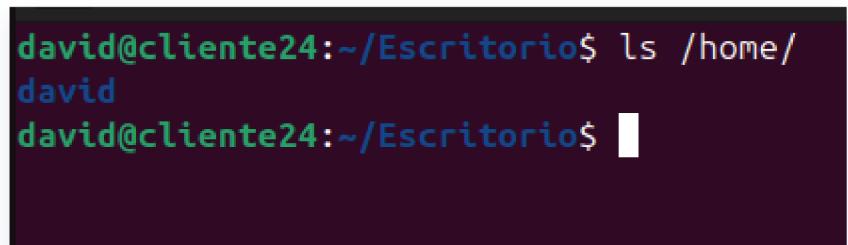
Recordar salir del modo `root` con el comando `exit` cuando no haya que hacer tareas con privilegios de administración.

```
# 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.

Para cada usuario se crea un directorio dentro de `/home`. Es el directorio de trabajo:

```
$ls /home/  
alumno ana berta ftp laura luisjose pedro
```



```
david@cliente24:~/Escritorio$ ls /home/  
david  
david@cliente24:~/Escritorio$
```

Cambio de grupo y de dueño (chown, chgrp)

Imaginemos que el fichero `informe.txt` ha sido creado por el usuario `pedro`. Por defecto, el dueño de un archivo es el usuario que lo crea, en este caso `pedro`. El grupo del usuario `pedro`, como hemos visto antes es `oficina_malaga`.

```
$ su pedro
$ cd
$ pwd
/home/pedro
$ touch informe.txt
$ ls -l
-rw-r--r-- 1 pedro oficina_malaga 0 2009-03-19 12:46 informe.txt
```

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

```
# mv informe.txt /home/laura/
# cd /home/laura/
# chown laura informe.txt
# ls -l
-rw-r--r-- 1 laura oficina_malaga 0 2009-03-19 12:46 informe.txt
```

Ahora el fichero tiene al usuario `laura` como propietario.

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

Cambio de privilegios (chmod)

`chmod` sirve para cambiar los privilegios de uno o varios ficheros. Los mismos permisos que se pueden ver con `ls -l`

```
$ ls -l
-rw-r--r-- 1 pedro oficina_malaga 0 2009-03-19 15:38 hola_mundo.rb
$ chmod +x hola_mundo.rb
$ ls -l
-rwxr-xr-x 1 pedro oficina_malaga 0 2009-03-19 15:38 hola_mundo.rb
```

Hemos añadido el permiso de ejecución al fichero `hola_mundo.rb`. Ahora hay tres `x` la que corresponde al dueño del fichero, la de todos los usuarios que pertenecen al grupo y la del resto de usuarios.

Si no se especifica la letra, pertenece a los usuarios(u, g, o). Se pueden indicar de forma explícita con el carácter a (all).

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

Quitaremos ahora el permiso de ejecucion para el resto de users(others) y daremos permiso de escritura(write) a los usuarios del mismo grupo(group).

```
$ ls -l
-rwxr-xr-x 1 pedro oficina_malaga 0 2009-03-19 15:38 hola_mundo.rb
$ chmod o-x hola_mundo.rb
$ chmod g+w hola_mundo.rb
$ ls -l
-rwxrwxr-- 1 pedro oficina_malaga 0 2009-03-19 15:38 hola_mundo.rb
```

Método simbólico: método que usa los caracteres `rwx`.

4 2 1

Total

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

Esta línea:

```
$ chmod 755 hola_mundo.rb
```

sería equivalente a estas tres

```
$ chmod u+rwx hola_mundo.rb
$ chmod g+rx-w hola_mundo.rb
$ chmod o+rx-w hola_mundo.rb
```

Resumen capítulo 4

<i>Comando</i>	<i>Acción</i>
ls -l	Muestra, entre otras cosas, información sobre los permisos, el usuario y el grupo al que pertenece el fichero.
sudo	Permite ejecutar comandos como root.
su	Cambia de usuario.
whoami	Muestra el nombre del usuario actual.
groups	Muestra el/los grupos/s a los que pertenece el usuario actual.
groupadd	Añade un nuevo grupo.
groupdel	Borra un grupo.
groupmod	Modifica las características de un grupo.
adduser	Añade un nuevo usuario.
userdel	Borra un usuario.
usermod	Modifica las características de un usuario.
passwd	Asigna o cambia la clave de un usuario.
chown	Cambia el dueño de un archivo.
chgrp	Cambia el grupo al que pertenece un archivo.
chmod	Cambia los permisos.

1. Completa la siguiente tabla:

654	rw-r-xr--
766	rwxrw-rw
777	rwxrwxrwx
520	r-x-w----
764	rwxrw-r--
440	r--r----

2. Crea los grupos oficina1 y oficina2.

```
# groupadd oficina1  
# groupadd oficina2
```

3. Crea los usuarios paco y pablo. Estos usuarios deben pertenecer únicamente al grupo oficina1.

```
# adduser paco --ingroup oficina1  
# adduser pablo --ingroup oficina1
```

4. Crea los usuarios alba y nerea. Estos usuarios deben pertenecer únicamente al grupo efficient.

```
# adduser alba --ingroup oficina2  
# adduser nerea --ingroup oficina2
```

5. Como usuario paco Crea un fichero con nombre topsecret.txt en su directorio de trabajo al que únicamente él tenga acceso, tanto de lectura como de escritura.

```
$ su paco  
$ cd  
$ touch top_secret.txt  
$ chmod 600 top_secret.txt
```

6. Crea otro fichero, también como usuario paco, con nombre ventas_trimestre.txt al que tengan acceso, tanto para leer como para escribir todos los usuarios que pertenezcan al mismo grupo. Se deben dejar los permisos que haya por defecto para el dueño y para el resto de usuarios. Comprueba como usuario pablo que puedes modificar el fichero.

```
$ touch ventas_trimestre.txt  
$ chmod g+rw ventas_trimestre.txt  
Comprobamos que el usuario pablo puede modificar este fichero, ya que es  
miembro del mismo  
grupo:  
$ exit  
$ su pablo  
$ vi /home/paco/ventas_trimestre.txt
```

7. Como usuario alba, crea un fichero con nombre empleados.txt al que pueda acceder cualquier usuario para leer su contenido, y cualquier usuario del mismo grupo para leer o escribir.

```
$ exit  
$ su alba  
$ cd  
$ touch empleados.txt  
$ chmod 664 empleados.txt
```

8. Copia el fichero empleados.txt al directorio de trabajo de alumno (crea también el usuario alumno si no está creado). Cambia el propietario y el grupo al que pertenece el fichero, ahora debe ser alumno.

```
$ exit  
$ sudo cp /home/alba/empleados.txt /home/alumno/  
$ sudo chown alumno /home/alumno/empleados.txt  
$ sudo chgrp alumno /home/alumno/empleados.txt
```

9. Como usuario pablo, copia un programa del directorio /usr/bin al directorio de trabajo con un nombre diferente. Por ejemplo xclock se puede copiar como reloj. Mira los permisos de este programa. Comprueba que se puede ejecutar. Puede que sea necesario dar permiso para que otros usuarios distintos al actual puedan ejecutar aplicaciones en el entorno gráfico, basta con ejecutar como administrador: xhost +.

```
luisjose@luisjose-xps1330:~$ su pablo
Contraseña:
pablo@luisjose-xps1330:/home/luisjose$ cd
pablo@luisjose-xps1330:~$ cp /usr/bin/xclock reloj
pablo@luisjose-xps1330:~$ ls -l
total 32
lrwxrwxrwx 1 pablo oficina1 26 2009-03-23 10:28 Examples →
/usr/share/example-content
-rwxr-xr-x 1 pablo oficina1 32568 2009-03-23 11:18 reloj
pablo@luisjose-xps1330:~$ ./reloj
No protocol specified
Error: Can't open display: :0.0
```

10. Cambia los permisos de reloj de tal forma que sólo lo pueda ejecutar el propietario del archivo.

```
$ chmod go-x reloj
```

11. Crea el usuario modesto, perteneciente a oficina2. Dentro de su directorio de trabajo, crea un directorio de nombre compartido_con_todos.

```
$ exit
$ sudo adduser modesto --ingroup oficina2
$ su modesto
$ cd
$ mkdir compartido_con_todos
```

12. Cambia de usuario en el entorno gráfico (botón salir y botón cambiar de usuario) y entra como modesto. Crea con OpenOffice.org Calc los ficheros telefono_contactos.ods, gastos_marzo.ods y sueldos.ods. Inserta varias entradas en cada uno de los ficheros y grábalo todo en el directorio compartido_con_todos.

13. Da permiso de lectura a la carpeta compartido_con_todos y a todos los ficheros que contenga para todos los usuarios.

```
chmod -R a+r compartido_con_todos
```

14. Restringe el acceso de escritura sobre el fichero telefono_contactos para que sólo lo puedan modificar los usuarios del grupo al que pertenece su propietario.

```
cd compartido_con.todos  
chmod g+w telefono_contactos.ods  
chmod o-w telefono_contactos.ods (en realidad esta línea sería redundante)
```

15. Cambia los permisos de gastos_marzo para que sólo pueda modificarlo su propietario y leerlo cualquiera del mismo grupo.

```
chmod 640 gastos_marzo.ods
```

16. Cambia los permisos de sueldos para que sólo su dueño tenga acceso a él, tanto para lectura como para escritura.

```
chmod 600 sueldos.ods
```

17. 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.

No. Un usuario que no tenga -r no puede acceder a los ficheros de ese directorio.

Comandos de Red en Linux

1. ip a

El comando `ip a` o `ip address` muestra la configuración de las interfaces de red y las direcciones IP asociadas. Alternativa a `ipconfig`. Nos da detalles sobre el estado de las interfaces, direcciones IP asignadas, máscaras de red, etc.

```
ip a
```

2. ifconfig

Muestra y configura las interfaces de red. Se está reemplazando por `ip`. Permite ver las direcciones IP asignadas y configurar las interfaces.

```
ifconfig
```

3. ping

Verifica la conectividad de red entre dos dispositivos mediante el envío de paquetes ICMP. Mide la latencia y pérdida de paquetes entre el host de origen y el destino.

```
ping [dirección IP o dominio]
```

4. nslookup

Se usa para realizar consultas DNS, lo que permite resolver nombres de dominio a direcciones IP o viceversa. Es útil para verificar la configuración DNS y disponibilidad de nombres de dominio.

```
nslookup [nombre de dominio o dirección IP]
```

5. netstat

Proporciona estadísticas detalladas sobre las conexiones de red, incluyendo las conexiones activas, tablas de enrutamiento y estadísticas de las interfaces. Es útil para diagnosticar problemas de red y monitorear el tráfico de red.

```
netstat
```

6. curl

Es una herramienta de línea de comandos para transferir datos usando varios protocolos, como HTTP, FTP, y más. Se usa frecuentemente para descargar archivos, hacer pruebas de APIs, y verificar la conectividad de servicios web.

```
curl [URL]
```

7. hostname

Muestra o configura el nombre del host del sistema. Es útil para verificar el nombre de la máquina o cambiarlo temporalmente sin reiniciar el sistema.

```
hostname
```

8. whois

Realiza consultas sobre la información de registro de un dominio, incluyendo datos del propietario, fechas de creación y expiración, y los servidores DNS asociados. Es útil para obtener información detallada de dominios en internet.

```
whois [nombre de dominio]
```

9. ip route

Sirve para **ver, añadir, modificar y borrar rutas** en la tabla de enrutamiento del kernel

Se usa principalmente para:

- Mostrar rutas configuradas.
- Definir rutas hacia redes y hosts.
- Configurar el gateway (ruta por defecto).
- Eliminar rutas.

1. Mostraremos la tabla de enrutamiento.

```
ip route show
```

Ejemplo de salida:

```
default via 192.168.1.1 dev eth0 proto dhcp metric 100 192.168.1.0/24 dev eth0
proto kernel scope link src 192.168.1.50 metric 100
```

Aquí vemos que la ruta por defecto (`default`) va a través de `192.168.1.1` en `eth0` . Y que la red local `192.168.1.0/24` está directamente conectada.

2. Añadir una ruta a una red

```
sudo ip route add 10.10.10.0/24 via 192.168.1.1 dev eth0
```

Para llegar a la red `10.10.10.0/24` , se usará la puerta de enlace `192.168.1.1` a través de `eth0` .

3. Añadir una ruta a un host específico.

```
sudo ip route add 172.16.0.50 via 192.168.1.1
```

Sólo el host `172.16.0.50` pasará por el gateway `192.168.1.1`

4. Añadir la ruta por defecto(gateway)

```
sudo ip route add default via 192.168.1.1 dev eth0
```

Configura el gateway predeterminado para todo el tráfico no especificado.

5. Eliminar una ruta

```
sudo ip route del 10.10.10.0/24
```

Elimina la ruta hacia la red `10.10.10.0/24` .

6. Reemplazar una ruta

```
sudo ip route replace default via 192.168.1.254 dev eth0
```

Si existe una ruta por defecto la sobrescribe; si no, la añade.

7. Rutas avanzadas (por tabla de enrutamiento)

```
ip route show table all
```

Permite ver todas las tablas de enrutamiento, no sólo la principal (útil en *policy routing*).

10. ip link

Sirve para *mostrar y gestionar interfaces de red* en Linux (`eth0`, `wlan0`, `lo` , etc)

Con él puedes:

- Ver información detallada de las interfaces.
- Activar o desactivar una interfaz (UP/DOWN).
- Cambiar nombre a una interfaz.
- Cambiar dirección MAC.
- Ajustar parámetros avanzados de red.

Ejemplos de uso

1. Mostrar todas las interfaces de red

```
ip link show
```

Salida típica:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode
DEFAULT group default qlen 1000
    link/ether 08:00:27:4a:35:9c brd ff:ff:ff:ff:ff:ff
```

Te muestra *ID de interfaz, nombre, estado, MTU, dirección MAC, etc.*

2. Mostrar sólo una interfaz concreta.

```
ip link show eth0
```

Filtra la información de la interfaz `eth0`

3. Activar una interfaz

```
sudo ip link set eth0 up
```

Activa la tarjeta de red `eth0`.

4. Desactivar una interfaz

```
sudo ip link set eth0 down
```

Desactiva la interfaz `eth0` (útil para pruebas o evitar tráfico temporalmente).

5. Cambiar el nombre de una interfaz

```
sudo ip link set eth0 name lan0
```

Renombra la interfaz de `eth0` a `lan0`

6. Cambiar la dirección MAC

```
sudo ip link set dev eth0 address 02:1A:2B:3C:4D:5E
```

Establece una nueva dirección MAC para la interfaz `eth0`.(Ojo: algunas tarjetas no permiten cambiar la MAC).

CUESTIONES

1. Muestra todas las interfaces de red activas y sus direcciones IP en el sistema.

```
ip a
```

2. ¿Cómo mostrarías solo la información de la interfaz de red eth0 usando ip a?

```
ip a show eth0
```

3. Configura manualmente la dirección IP 192.168.1.100/24 en la interfaz eth0 con ifconfig
4. Envía 10 paquetes ICMP a la dirección IP 8.8.8.8 usando ping.

```
ping -c 10 8.8.8.8
```

5. Consulta la dirección IP de www.example.com usando nslookup.
6. Muestra las conexiones TCP activas en el sistema usando netstat.

```
netstat -tn
```

7. Descarga el contenido de la página principal de www.example.com usando curl y guárdalo en un archivo llamado example.html.
8. Consulta el nombre del host actual del sistema.

```
hostname
```

9. Obtén la información de registro del dominio example.com usando whois.
10. Cambia temporalmente el nombre del host a servidor01 usando hostname.

```
sudo hostname servidor01
```

11. Envía un ping a la dirección 192.168.1.1 y muéstralos en modo detallado (verbose).

```
ping -v 192.168.1.1
```

12. Muestra las estadísticas de la red, como la cantidad de paquetes transmitidos, usandonetstat.
13. Realiza una consulta inversa para obtener el nombre de dominio asociado a la IP 8.8.8.8 connslookup.
14. Configura temporalmente la máscara de subred 255.255.255.128 en la interfaz eth1 usandoifconfig.

```
sudo ifconfig eth1 192.168.1.100 netmask 255.255.255.128 up
```

15. Muestra las rutas de enrutamiento actuales usando netstat.

```
netstat -rn
```

16. Realiza una solicitud HTTP GET a la API de GitHub para obtener los repositorios de usuario123usando curl.
17. Envía un ping a la dirección 2001:4860:4860::8888 (IPv6 de Google) con ping6 y limita los paquetes a 4.

```
ping6 -c 4 2001:4860:4860 :: 8888
```

18. Obtén las estadísticas de los sockets activos en el sistema con netstat.
19. Cambia temporalmente la dirección MAC de la interfaz eth0 a 00:11:22:33:44:55 usandoifconfig.
20. Realiza una solicitud HTTP POST a <https://httpbin.org/post> enviando el usuario admin y lacontraseña 12345 usando curl.
21. Consulta el nombre de dominio completo (FQDN) de tu sistema usando hostname.

```
hostname -f
```

22. Muestra solo las conexiones activas en la interfaz eth0 usando netstat.

```
netstat -i eth0
```

23. Muestra las conexiones activas con nombres de dominio en lugar de direcciones IP usandonetstat.

24. Configura una nueva puerta de enlace predeterminada con la dirección 192.168.1.1 usandoip route.

25. ¿Qué comando usarías para ver todas las rutas configuradas en tu sistema?

26. ¿Cómo configuras que todo el tráfico destinado a la red 10.10.10.0/24 pase por el gateway192.168.1.1 en la interfaz eth0?

```
sudo ip route add 10.10.10.0/24 via 192.168.1.1 dev eth0
```

27. ¿Cómo eliminas la ruta añadida en el ejercicio anterior?

```
sudo ip route del 10.10.10.0/24
```

28. Si la interfaz eth0 está deshabilitada, ¿qué comando usarías para levantarla?

29. ¿Qué comando utilizas para asignar la dirección MAC 02:1A:2B:3C:4D:5E a la interfaz eth0?

30. ¿Cómo renombrarías la interfaz eth0 para que pase a llamarse lan0?

```
sudo ip link set eth0 down  
sudo ip link set eth0 name lan0  
sudo ip link set lan0 up
```