

UD 5. Gestión del almacenamiento en Linux.

Starting X display
GNU/Linux 3.1
m: _

UD 5 – Gestión del almacenamiento en Linux

INDICE DE CONTENIDOS

1. La shell de Linux

2. Usuarios básicos en un sistema Linux

3. Estructura del sistema de ficheros de Linux

4. La gestión de ficheros

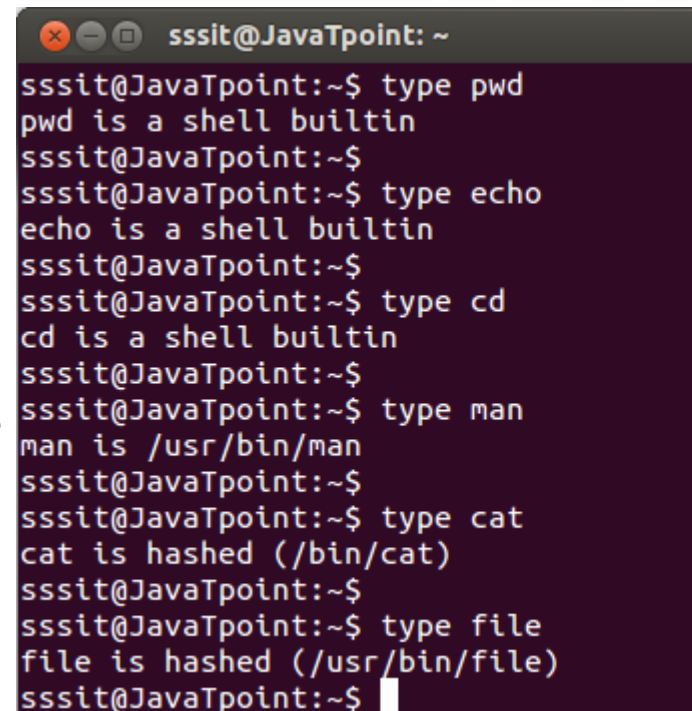
UD 5 – Gestión del almacenamiento en Linux

1. La shell de Linux

Es un intérprete de comandos (tipo de programa) que permite al usuario interactuar con el núcleo del sistema mediante órdenes que se escriben por teclado. También se le llama *terminal*.

La forma más rápida de abrir una ventana del terminal en Linux es usando las teclas Ctrl + Alt + T.

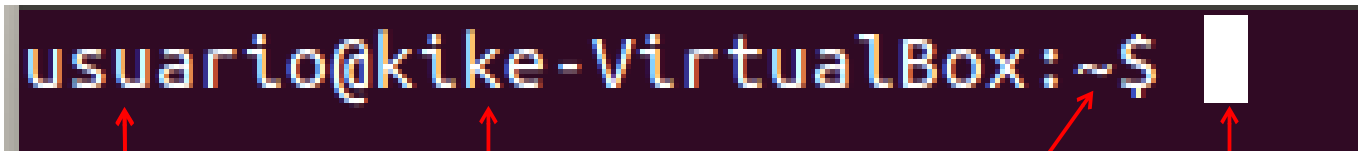
Existen varias *shells* siendo posible intercambiarlas (*bash*, *sh*, *dash*...) aunque se parecen bastante entre ellas.



```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ type pwd  
pwd is a shell builtin  
sssit@JavaTpoint:~$  
sssit@JavaTpoint:~$ type echo  
echo is a shell builtin  
sssit@JavaTpoint:~$  
sssit@JavaTpoint:~$ type cd  
cd is a shell builtin  
sssit@JavaTpoint:~$  
sssit@JavaTpoint:~$ type man  
man is /usr/bin/man  
sssit@JavaTpoint:~$  
sssit@JavaTpoint:~$ type cat  
cat is hashed (/bin/cat)  
sssit@JavaTpoint:~$  
sssit@JavaTpoint:~$ type file  
file is hashed (/usr/bin/file)  
sssit@JavaTpoint:~$
```

UD 5 – Gestión del almacenamiento en Linux

Cuando abrimos la shell, el sistema nos “saluda” con un indicador llamado ***prompt***. Veamos qué pinta tiene:

A screenshot of a Linux terminal window with a dark purple background. The prompt text is 'usuario@kike-VirtualBox:~\$' followed by a white cursor. Red arrows point from the text below to specific parts of the prompt: one to 'usuario', one to '@', one to 'kike-VirtualBox', one to ':', one to '~', one to '\$', and one to the cursor.

Nombre de usuario @ nombre máquina : directorio actual \$ cursor escritura

Separadores

Aquí podemos escribir comandos para interactuar con el sistema. Cuando lanzamos la ejecución de un comando sabremos que ha terminado cuando vuelva a salir el prompt.

OJO: Linux es “sensible” a mayúsculas y minúsculas. Por ejemplos: Leeme.txt y leeme.txt son dos ficheros distintos.

UD 5 – Gestión del almacenamiento en Linux

SINTAXIS DE LOS COMANDOS DE LINUX

Las órdenes o comandos de Linux tienen la siguiente sintaxis:

```
comando [-modificadores] [parámetro1]  
[parámetro2] [...]
```

Todo que está entre corchetes es opcional, es decir, podría omitirse.

Los modificadores sirven para “cambiar” el comportamiento del comando y van precedidos de un guión “-”.

Los parámetros serán necesarios o no dependiendo del comando. Por ejemplo, para copiar un fichero a una carpeta necesitaremos el nombre del fichero (*parámetro1*) y la carpeta en la que lo copiaremos (*parámetro2*).

Ej: `cp -R /home/pepito/ /mnt/usb/` (copia recursivamente)

UD 5 – Gestión del almacenamiento en Linux

Algunos comandos sencillos:

echo: escribe en pantalla una frase o el valor de una variable del sistema. Ejemplos.

echo Esto es una frase

echo \$SHELL (nos indica qué shell estamos usando)

clear: limpia el terminal

man: muestra el manual de uso de un comando. Ej:

man date (nos muestra el manual del comando date)

exit: cierra la aplicación

Navegación por los comandos: si pulsamos las teclas de arriba y abajo de los cursores “navegamos” por el historial de comandos
tecleados

Haz pruebas de estos comandos en la shell

UD 5 – Gestión del almacenamiento en Linux

INDICE DE CONTENIDOS

1.La shell de Linux

2.**Usuarios básicos en un sistema Linux**

3.Estructura del sistema de ficheros de Linux

4.La gestión de ficheros

UD 5 – Gestión del almacenamiento en Linux

2. Usuarios básicos en un sistema Linux

Para Linux hay tres tipos de usuarios:

- Usuario root
- Usuarios normales o finales
- Usuario especiales o de sistema

Root es el “dueño” (owner) del sistema y tiene plenos poderes sobre él. También se le llama **administrador** o **superusuario**.

Es el único que puede administrar las cuentas de los usuarios.



UD 5 – Gestión del almacenamiento en Linux

Los **usuarios normales o finales** son los que pueden utilizar las personas que se conectan al sistema. Este tipo de usuarios se caracteriza por:

- Normalmente poseen una **carpeta propia de trabajo** (home) en la que pueden crear sus propios archivos.
- Se **organizan por “grupos de usuarios”**.
- Un usuario siempre debe pertenecer, **al menos, a 1 grupo** de usuarios (su grupo primario)

Los **usuarios especiales o de sistema** son usuarios que crea el propio Linux para administrar sus “demonios” (servicios). Son usuarios que no pueden iniciar sesión y que son manejados por el sistema. **Ejemplos:** apache, daemon, bin, adm, lp, shutdown...



UD 5 – Gestión del almacenamiento en Linux

EL COMANDO SUDO

Cuando usamos usuarios finales (kike, pedro, juan...) no tenemos plenos poderes sobre el sistema, solo sobre los archivos de nuestra carpeta *home*.

Por otro lado, cuando instalamos un Ubuntu, nos pide que creemos nuestro usuario/contraseña inicial y el sistema nos crea un **usuario normal o final**.

Esto se hace así por razones de seguridad ante posibles ataques. Pero entonces **¿cómo puedo controlar el sistema si no soy root y nadie me ha dicho la contraseña del root?**



UD 5 – Gestión del almacenamiento en Linux

En Linux no son tontos y te dejan un camino libre. Resulta que nuestro usuario normal pertenece a un **grupo especial llamado sudo** y lo podemos comprobar con el comando **groups** que nos dice a qué grupos pertenece nuestro usuario:

```
kike@kike-VirtualBox:/etc$ groups  
kike adm cdrom sudo dip plugdev lpadmin sambashare
```

Los usuarios miembros de este grupo (sudoers) se consideran usuarios de confianza para el administrador y **se les permite realizar acciones propias de root puntualmente**.

Ahora, si queremos hacer algo propio del root, simplemente anteponemos la palabra **sudo** y podremos hacerlo. Veamos un ejemplo:



UD 5 – Gestión del almacenamiento en Linux

```
kike@kike-VirtualBox:/etc$ useradd pepito
useradd: Permission denied.
useradd: no se pudo bloquear /etc/passwd, inténtelo de nuevo.
kike@kike-VirtualBox:/etc$ sudo useradd pepito
[sudo] contraseña para kike:
kike@kike-VirtualBox:/etc$
```

useradd permite crear un usuario nuevo en el sistema, pero para realizarlo necesitamos privilegios de administrador.

Si anteponemos el comando **sudo**, nos pedirá nuestra contraseña (no la de root) y nos dejará hacerlo.

Esto tiene **muchas ventajas de seguridad**:

- root se usa lo menos posible (menos exposición)
- Se puede configurar qué permisos de administrador se le dan a los usuario *sudoers* (/etc/sudoers).
- El uso de **sudo** deja un rastro (log) que el administrador puede consultar



UD 5 – Gestión del almacenamiento en Linux

INDICE DE CONTENIDOS

1.La shell de Linux

2.Usuarios básicos en un sistema Linux

3.**Estructura del sistema de ficheros de Linux**

4.La gestión de ficheros

UD 5 – Gestión del almacenamiento en Linux

3. Estructura del sistema de ficheros de Linux

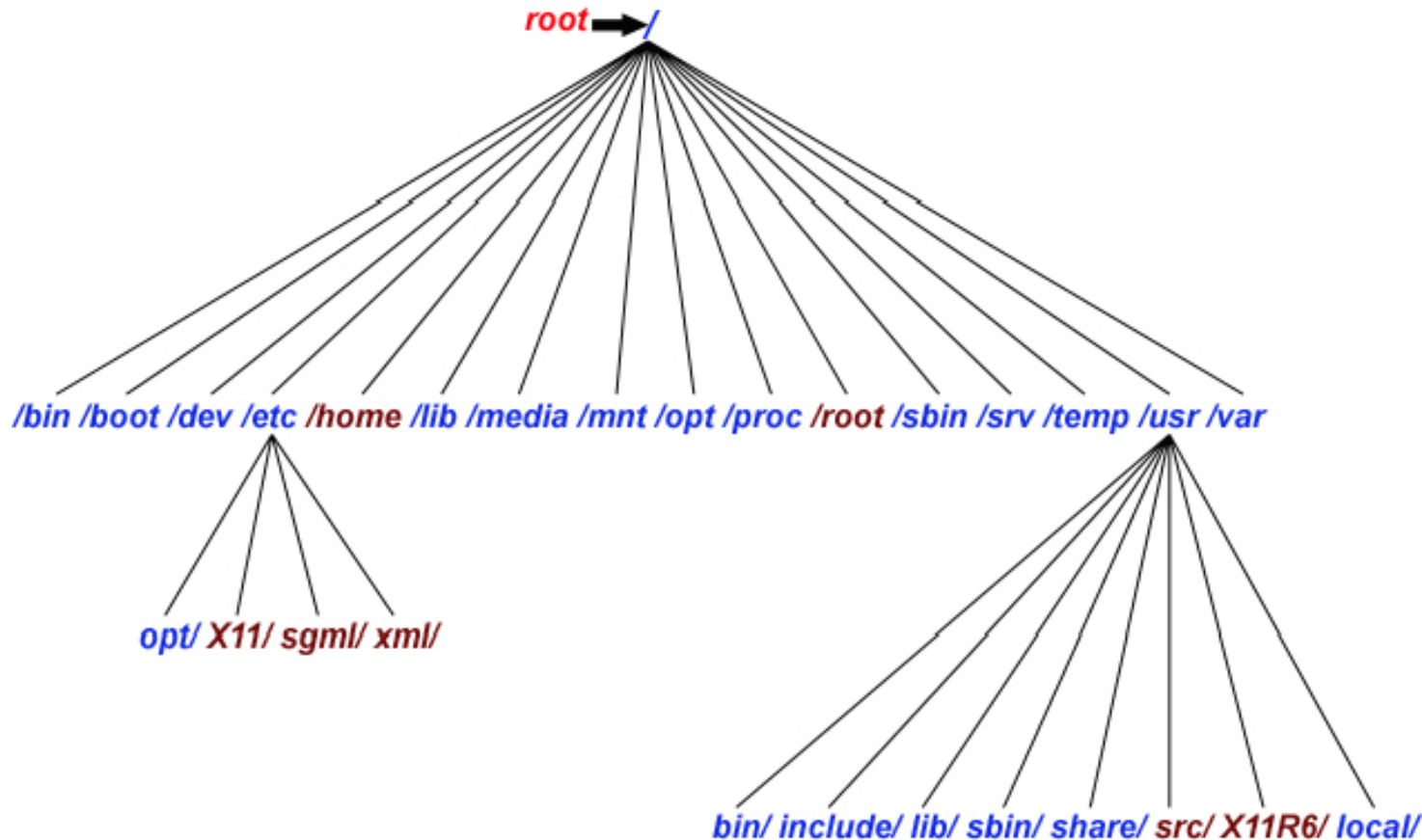
Linux es un sistema orientado a ficheros, es decir representa los distintos objetos del sistema (procesos, dispositivos, usuarios...) como si fueran ficheros o directorios. Aunque esto puede ser un poco raro al principio, es muy flexible.

Un sistema de ficheros se encarga de organizar la información del disco duro para ser usado de forma eficiente. Linux ha pasado por distintos tipos de sistemas de ficheros: EXT2, EXT3 y EXT4 (los dos últimos usan ***journaling*** para recuperarse ante fallos)



UD 5 – Gestión del almacenamiento en Linux

Linux guarda los ficheros y carpetas en una estructura en árbol donde el directorio superior (también llamado *raíz* o */*) es el padre de otras carpetas y éstas a su vez pueden ser padre de otras carpetas o ficheros.



UD 5 – Gestión del almacenamiento en Linux

Veamos algunas de las carpetas principales del sistema:

/bin: (binaries) almacena los comandos básicos del sistema (echo, date, ping, ls, cp...)

/sbin: (superuser binaries) almacena los comandos de administración del sistema (ifconfig, shutdown, fdisk, iptables...)

/home: contiene las carpetas personales de los usuarios estándar del sistema.

/root: es la carpeta personal del administrador del sistema.

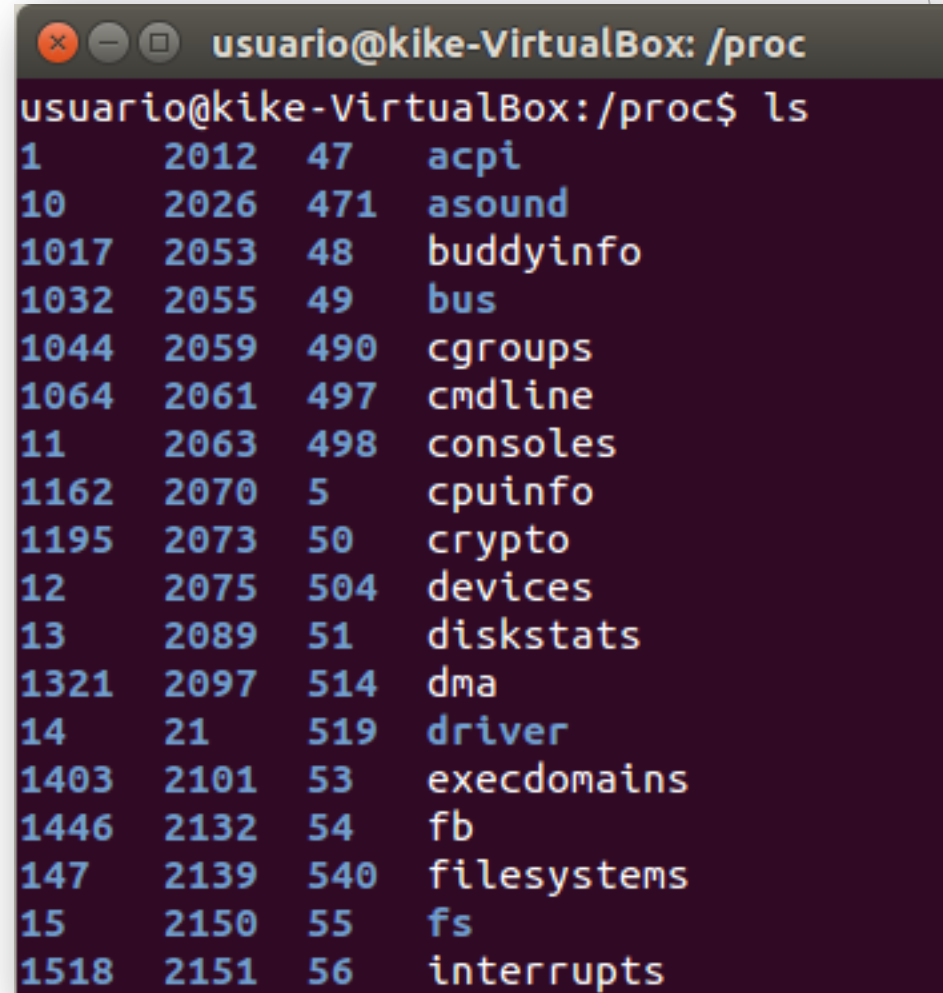
/etc: almacena los archivos de configuración del sistema y de aplicaciones importantes.

/usr: (user) almacena las aplicaciones de usuarios estándar

UD 5 – Gestión del almacenamiento en Linux

/proc: (process) esta carpeta contiene unos ficheros “virtuales” que representan a los distintos procesos que se están ejecutando en el sistema.

Podría decirse que es la versión en ficheros del “Administrador de tareas o procesos” del sistema



A terminal window titled 'usuario@kike-VirtualBox: /proc' showing the output of the 'ls' command. The output lists various system files and directories in the /proc filesystem, each preceded by a four-digit ID number. The files listed are: acpi, asound, buddyinfo, bus, cgroups, cmdline, consoles, cpuinfo, crypto, devices, diskstats, dma, driver, execdomains, fb, filesystems, fs, and interrupts.

```
usuario@kike-VirtualBox: /proc$ ls
1      2012  47    acpi
10     2026  471   asound
1017   2053  48    buddyinfo
1032   2055  49    bus
1044   2059  490   cgroups
1064   2061  497   cmdline
11     2063  498   consoles
1162   2070  5     cpuinfo
1195   2073  50    crypto
12     2075  504   devices
13     2089  51    diskstats
1321   2097  514   dma
14     21    519   driver
1403   2101  53    execdomains
1446   2132  54    fb
147    2139  540   filesystems
15     2150  55    fs
1518   2151  56    interrupts
```

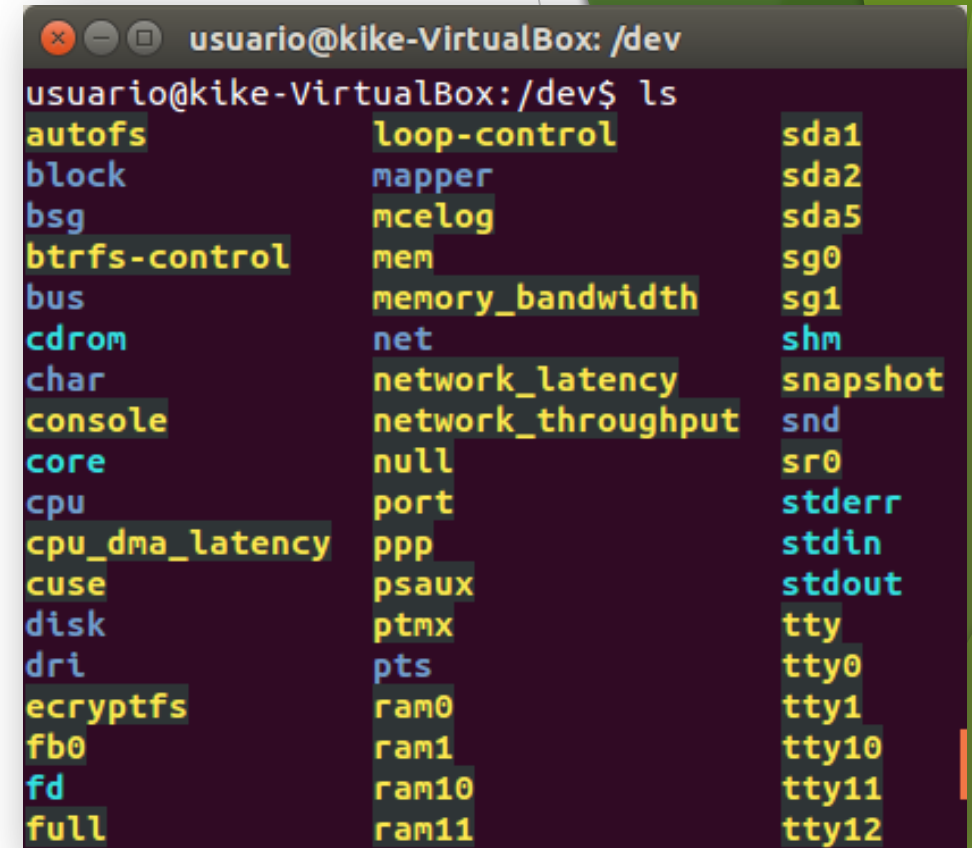
UD 5 – Gestión del almacenamiento en Linux

/dev: (devices) cada archivo de esta carpeta representa un dispositivo real o virtual del sistema. Ejemplos:

- **/dev/mem:** memoria
- **/dev/cpu:** la cpu
- **/dev/sda1:** (solid disc) la primera partición (1) del primer disco duro conectado al sistema (a)
- **/dev/sdc2:** (solid disc) la segunda partición (2) del tercer disco conectado al sistema (c)

Para saber más:

<https://techsolutionscenter.wordpress.com/2011/10/21/explicaciones-acerca-del-directorio-dev-en-sistemas-linux/>



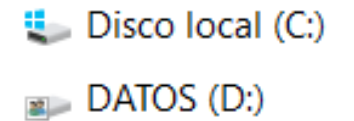
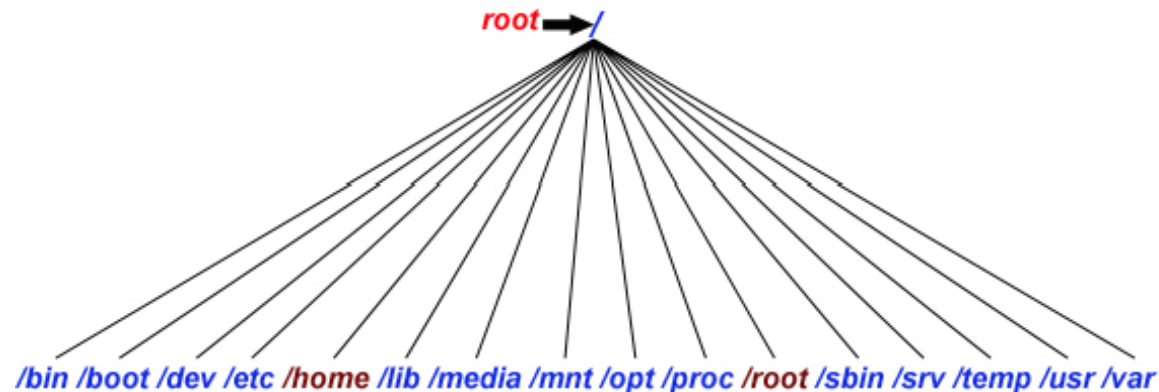
```
usuario@kike-VirtualBox: /dev
usuario@kike-VirtualBox: /dev$ ls
autofs          loop-control    sda1
block           mapper          sda2
bsg             mcelog          sda5
btrfs-control   mem            sg0
bus             memory_bandwidth sg1
cdrom           net             shm
char            network_latency snapshot
console         network_throughput snd
core           null            sr0
cpu             port            stderr
cpu_dma_latency ppp             stdin
cuse           psaux           stdout
disk           ptmx            tty
dri            pts             tty0
ecryptfs       ram0            tty1
fb0            ram1            tty10
fd             ram10           tty11
full           ram11           tty12
```

UD 5 – Gestión del almacenamiento en Linux

Dispositivos y puntos de montaje

En Windows, cada vez que conectamos un nuevo dispositivo que contenga un sistema de ficheros (pendrive, disco duro...) el sistema operativo lo identifica mediante una letra (D: E: F: ...) que nos permite acceder a este sistema de ficheros. A esta “letra” se le llama “unidad”.

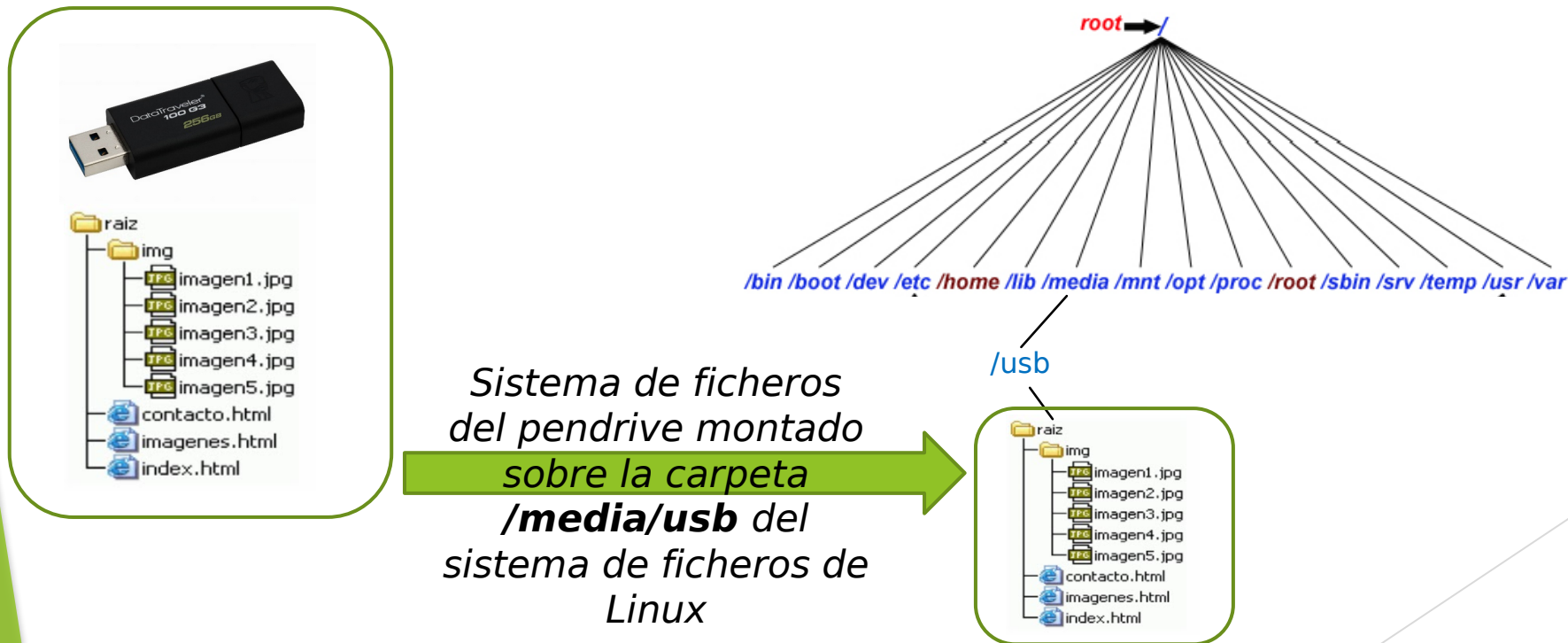
Sin embargo, en Linux NO EXISTEN la “unidades”. Para Linux solo existe **1 sistema de ficheros** que empieza en el nodo raíz / y todos los elementos “cuelgan” de este nodo.



UD 5 – Gestión del almacenamiento en Linux

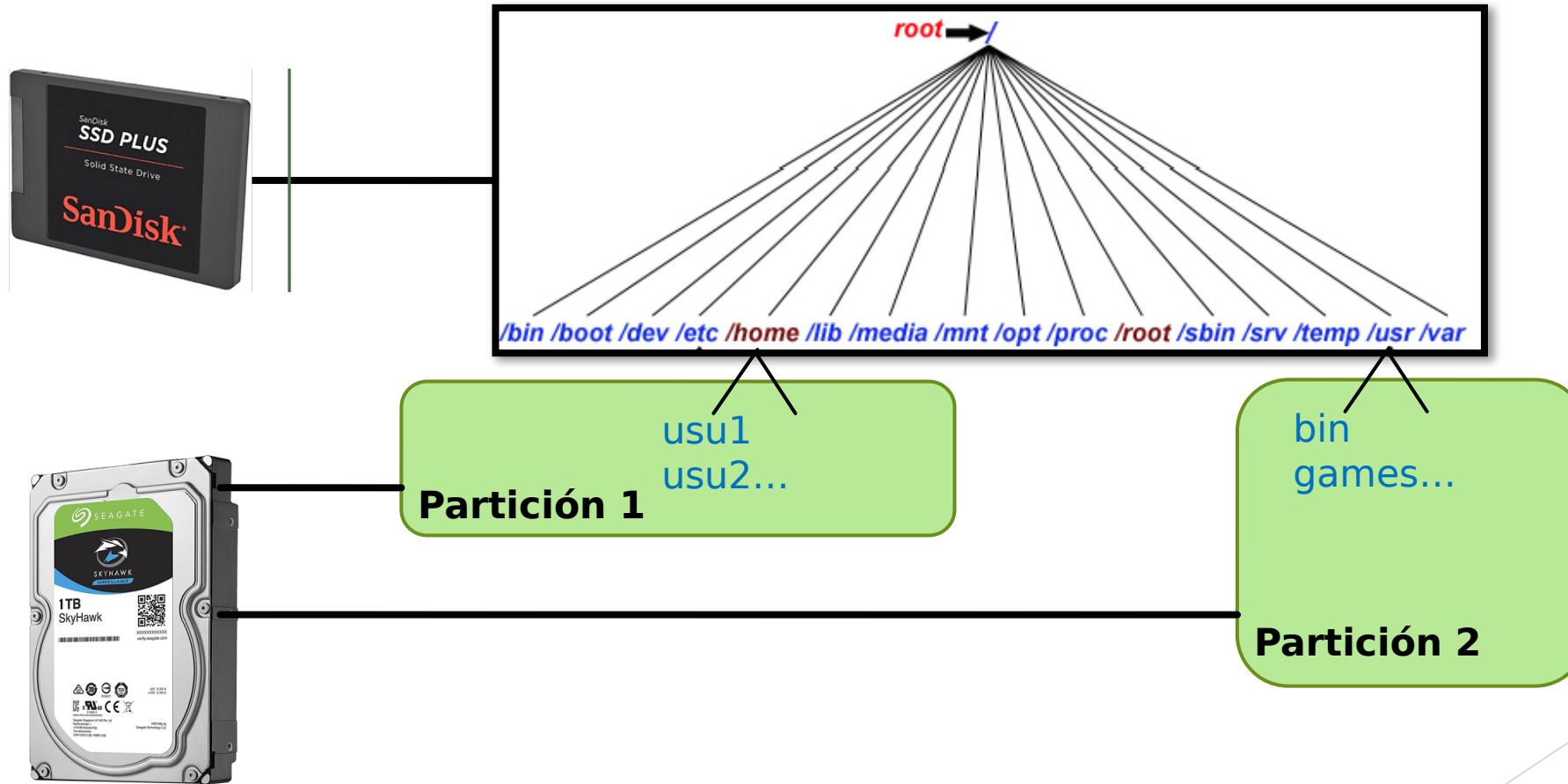
Entonces ¿cómo se **accede a los ficheros almacenados en un pen-drive** o una partición de un disco duro?

En Ubuntu Desktop cada vez que se conecta un dispositivo al sistema se “monta” sobre el sistema de ficheros raíz */*, es decir se le asigna una carpeta desde la que poder acceder al dispositivo. A esta carpeta se le llama “punto de montaje”



UD 5 – Gestión del almacenamiento en Linux

Este sistema es muy versátil y permite utilizar distintos discos duros y particiones para guardar distintos elementos del sistema:



UD 5 – Gestión del almacenamiento en Linux

INDICE DE CONTENIDOS

1.La shell de Linux

2.Usuarios básicos en un sistema Linux

3.Estructura del sistema de ficheros de Linux

4.**La gestión de ficheros**

UD 5 – Gestión del almacenamiento en Linux

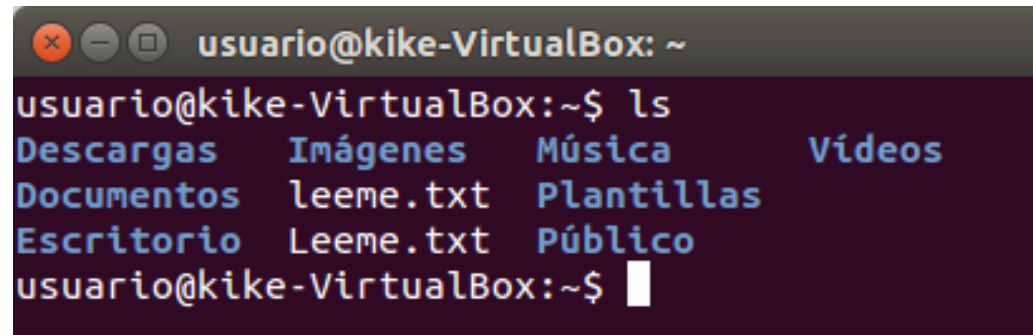
Veamos ahora los comandos que nos permiten manejarnos por el sistema de ficheros.

COMANDOS PARA VER Y MOVERSE POR LAS CARPETAS

ls (list): muestra el contenido de la carpeta actual.

En blanco → *Ficheros*

En azul → *Carpetas*



```
usuario@kike-VirtualBox: ~  
usuario@kike-VirtualBox:~$ ls  
Descargas  Imágenes  Música    Vídeos  
Documentos leeme.txt Plantillas  
Escritorio Leeme.txt Público  
usuario@kike-VirtualBox:~$
```

Si queremos seleccionar qué queremos listar podemos añadir comodines al comando ls:

***** : reemplaza a 0 o más caracteres (letras, dígitos, símbolos...)

? : reemplaza a 1 carácter

UD 5 – Gestión del almacenamiento en Linux

```
usuario@kike-VirtualBox: ~  
usuario@kike-VirtualBox:~$ ls  
Descargas  Escritorio  leeme.txt  Música     Público  
Documentos Imágenes   Leeme.txt  Plantillas Vídeos  
usuario@kike-VirtualBox:~$ ls *.txt  
leeme.txt  Leeme.txt  
usuario@kike-VirtualBox:~$
```

```
usuario@kike-VirtualBox: ~  
usuario@kike-VirtualBox:~$ ls  
Descargas  Escritorio  leeme.txt  Música     Público  
Documentos Imágenes   Leeme.txt  Plantillas Vídeos  
usuario@kike-VirtualBox:~$ ls ?eeme.txt  
leeme.txt  Leeme.txt  
usuario@kike-VirtualBox:~$
```

pwd (print working directory): muestra la ruta de la carpeta actual.

```
usuario@kike-VirtualBox: ~  
usuario@kike-VirtualBox:~$ pwd  
/home/usuario  
usuario@kike-VirtualBox:~$
```

UD 5 – Gestión del almacenamiento en Linux

cd (change directory): cambia la carpeta actual por otra. Ejemplo:

```
usuario@kike-VirtualBox: ~/Música
usuario@kike-VirtualBox:~$ pwd
/home/usuario
usuario@kike-VirtualBox:~$ cd Música
usuario@kike-VirtualBox:~/Música$ pwd
/home/usuario/Música
usuario@kike-VirtualBox:~/Música$
```

Tenemos dos formas de hacer el cambio de carpeta actual:

- ***Ruta absoluta***: se escribe la ruta hasta la nueva carpeta empezando en carpeta raíz /

```
usuario@kike-VirtualBox: /etc/X11
usuario@kike-VirtualBox:~$ pwd
/home/usuario
usuario@kike-VirtualBox:~$ cd /etc/X11
usuario@kike-VirtualBox:/etc/X11$ pwd
/etc/X11
usuario@kike-VirtualBox:/etc/X11$
```

UD 5 – Gestión del almacenamiento en Linux

- **Ruta relativa:** se escribe la ruta hasta la nueva carpeta empezando en carpeta actual. Para ello se usa dos comodines:
 - . se refiere a la carpeta actual (apenas se usa)

```
usuario@kike-VirtualBox: ~/Música
usuario@kike-VirtualBox:~$ pwd
/home/usuario
usuario@kike-VirtualBox:~$ cd ./Música
usuario@kike-VirtualBox:~/Música$
```

=

```
usuario@kike-VirtualBox: ~/Música
usuario@kike-VirtualBox:~$ pwd
/home/usuario
usuario@kike-VirtualBox:~$ cd Música
usuario@kike-VirtualBox:~/Música$
```

- .. se refiere a la carpeta padre de la actual (se usa mucho)

```
usuario@kike-VirtualBox: ~
usuario@kike-VirtualBox:~/Música$ pwd
/home/usuario/Música
usuario@kike-VirtualBox:~/Música$ cd ..
usuario@kike-VirtualBox:~$ pwd
/home/usuario
usuario@kike-VirtualBox:~$
```

```
usuario@kike-VirtualBox: ~/Videos
usuario@kike-VirtualBox:~/Música$ pwd
/home/usuario/Música
usuario@kike-VirtualBox:~/Música$ cd ../Videos
usuario@kike-VirtualBox:~/Videos$ pwd
/home/usuario/Videos
usuario@kike-VirtualBox:~/Videos$
```

UD 5 – Gestión del almacenamiento en Linux

COMANDOS PARA VER Y EDITAR FICHEROS DE TEXTO

touch: crea un fichero vacío con el nombre especificado. Ejemplo:

```
usuario@kike-VirtualBox: ~  
usuario@kike-VirtualBox:~$ ls  
Descargas  Escritorio  leeme.txt  Música     Público  
Documentos Imágenes   Leeme.txt  Plantillas Vídeos  
usuario@kike-VirtualBox:~$ touch patata.txt  
usuario@kike-VirtualBox:~$ ls  
Descargas  Imágenes  Música     Público  
Documentos leeme.txt patata.txt Vídeos  
Escritorio Leeme.txt Plantillas  
usuario@kike-VirtualBox:~$
```

nano: permite abrir y editar un fichero de texto.

Ejemplo:

```
usuario@kike-VirtualBox: ~  
usuario@kike-VirtualBox:~$ nano patata.txt
```

```
usuario@kike-VirtualBox: ~  
GNU nano 2.2.6          Archivo: patata.txt          Modificado  
Esto es una frase de ejemplo  
Y otra más...  
  
^G Ver ayuda  ^O Guardar    ^R Leer fich.^Y Pág. ant.  ^K Cortar Tex^C Posición  
^X Salir      ^J Justificar^W Buscar     ^V Pág. sig.  ^U PegarTxt   ^T Ortografía
```

Ctrl+o → Guardar

Ctrl+x → Salir

UD 5 – Gestión del almacenamiento en Linux

cat: permite visualizar el contenido de un fichero. Ejemplo:

```
usuario@kike-VirtualBox: ~  
usuario@kike-VirtualBox:~$ cat patata.txt  
Esto es una frase de ejemplo  
Y otra más...  
usuario@kike-VirtualBox:~$
```

more: permite visualizar el contenido de un fichero pero si el fichero es muy largo te presenta la información “en páginas”. Ejemplo:

```
usuario@kike-VirtualBox: ~  
usuario@kike-VirtualBox:~$ more /etc/profile
```

Con Q se sale de more

```
usuario@kike-VirtualBox: ~  
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))  
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).  
  
if [ "$PS1" ]; then  
  if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then  
    # The file bash.bashrc already sets the default PS1.  
    # PS1='\h:\w\$ '  
    if [ -f /etc/bash.bashrc ]; then  
      . /etc/bash.bashrc  
    fi  
  else  
    if [ "`id -u`" -eq 0 ]; then  
      PS1='# '  
    else  
      PS1='$ '  
    fi  
  fi  
fi  
  
# The default umask is now handled by pam_umask.  
# See pam_umask(8) and /etc/login.defs.  
  
if [ -d /etc/profile.d ]; then  
  --Más-- (85%)
```


UD 5 – Gestión del almacenamiento en Linux

head: permite ver las 10 primeras líneas de un fichero.



```
usuario@kike-VirtualBox: ~  
usuario@kike-VirtualBox:~$ head /etc/profile  
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))  
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).  
  
if [ "$PS1" ]; then  
    if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then  
        # The file bash.bashrc already sets the default PS1.  
        # PS1='\h:\w\$ '  
        if [ -f /etc/bash.bashrc ]; then  
            . /etc/bash.bashrc  
        fi  
    fi  
usuario@kike-VirtualBox:~$
```

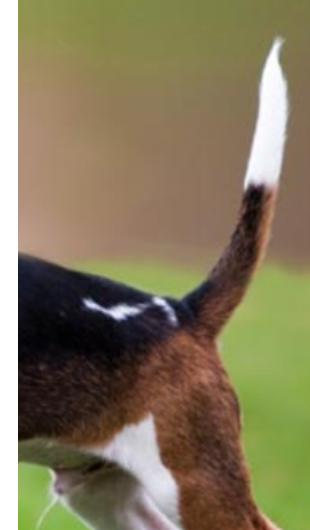
Si quieres ver *n* líneas de un fichero se puede indicar como parámetro:

```
usuario@kike-VirtualBox: ~  
usuario@kike-VirtualBox:~$ head -n 15 /etc/profile  
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))  
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).  
  
if [ "$PS1" ]; then  
    if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then  
        # The file bash.bashrc already sets the default PS1.  
        # PS1='\h:\w\$ '  
        if [ -f /etc/bash.bashrc ]; then  
            . /etc/bash.bashrc  
        fi  
    else  
        if [ "`id -u`" -eq 0 ]; then  
            PS1='# '  
        else  
            PS1='$ '  
        fi  
    fi  
usuario@kike-VirtualBox:~$
```


UD 5 – Gestión del almacenamiento en Linux

tail: permite ver las 10 últimas líneas de un fichero.
Funciona igual que head.

Estos dos comandos se suelen usar en ficheros de log para ver si ha ocurrido algo en las últimas o primeras líneas que sea de interés.



COMANDOS PARA CREAR Y BORRAR CARPETAS

mkdir (make directory): crear una o varias carpetas

```
usuario@mimaquina: ~  
usuario@mimaquina:~$ ls  
Descargas  Escritorio  leeme.txt  Música     Plantillas  Vídeos  
Documentos Imágenes   Leeme.txt  patata.txt Público  
usuario@mimaquina:~$ mkdir direc1 direct2  
usuario@mimaquina:~$ ls  
Descargas  direct2     Escritorio  leeme.txt  Música     Plantillas  
direc1     Documentos  Imágenes   Leeme.txt  patata.txt Público  
usuario@mimaquina:~$
```



UD 5 – Gestión del almacenamiento en Linux

rmdir (remove directory): borra uno o más directorios si están vacíos.

```
usuario@mimaquina: ~  
usuario@mimaquina:~$ ls  
Descargas  Documentos  leeme.txt  patata.txt  Videos  
direct1    Escritorio  Leeme.txt  Plantillas  
direct2    Imágenes   Música     Público  
usuario@mimaquina:~$ rmdir direct1 direct2  
usuario@mimaquina:~$ ls  
Descargas  Escritorio  leeme.txt  Música      Plantillas  Videos  
Documentos Imágenes   Leeme.txt  patata.txt  Público  
usuario@mimaquina:~$
```



COMANDOS PARA MANEJAR FICHEROS

cp (copy): permite copiar ficheros o carpetas desde uno o varios orígenes a un destino.

Veamos algunos ejemplos:



UD 5 – Gestión del almacenamiento en Linux

Copia el fichero
patata.txt a la carpeta
Descargas

```
usuario@mimaquina: ~
usuario@mimaquina:~$ ls
Descargas  Escritorio  leeme.txt  Música      Plantillas  Vídeos
Documentos Imágenes   Leeme.txt  patata.txt  Público
usuario@mimaquina:~$ cp patata.txt Descargas
usuario@mimaquina:~$
```

Copia los ficheros
leeme.txt y *Leeme.txt*
a la carpeta *Plantillas*

```
usuario@mimaquina: ~
usuario@mimaquina:~$ ls
Descargas  Escritorio  leeme.txt  Música      Plantillas  Vídeos
Documentos Imágenes   Leeme.txt  patata.txt  Público
usuario@mimaquina:~$ cp leeme.txt Leeme.txt Plantillas
usuario@mimaquina:~$
```

También podemos
usar comodines (* y ?)

```
usuario@mimaquina: ~
usuario@mimaquina:~$ ls
Descargas  Escritorio  leeme.txt  Música      Plantillas  Vídeos
Documentos Imágenes   Leeme.txt  patata.txt  Público
usuario@mimaquina:~$ cp *eeme.* Público
usuario@mimaquina:~$
```

Con la opción -r
copiamos carpetas
completas con todo su
contenido

```
usuario@mimaquina: ~
usuario@mimaquina:~$ ls
Descargas  Escritorio  leeme.txt  Música      Plantillas  Vídeos
Documentos Imágenes   Leeme.txt  patata.txt  Público
usuario@mimaquina:~$ cp -r Público Vídeos
usuario@mimaquina:~$
```

UD 5 – Gestión del almacenamiento en Linux

mv (move): tiene dos comportamientos:

Mueve uno o varios ficheros a un directorio

```
usuario@mimaquina: ~  
usuario@mimaquina:~$ ls  
Descargas  Escritorio  leeme.txt  Música      Plantillas  Vídeos  
Documentos  Imágenes   Leeme.txt  patata.txt  Público  
usuario@mimaquina:~$ mv patata.txt Escritorio
```



También podemos usar comodines (* y ?) para mover varios archivos de una vez

```
usuario@mimaquina: ~  
usuario@mimaquina:~$ ls  
Descargas  Escritorio  leeme.txt  Música      Público  
Documentos  Imágenes   Leeme.txt  Plantillas  Vídeos  
usuario@mimaquina:~$ mv *eeme.* Imágenes  
usuario@mimaquina:~$ ls  
Descargas  Escritorio  Música      Público  
Documentos  Imágenes   Plantillas  Vídeos  
usuario@mimaquina:~$
```

Cambia el nombre de un fichero o carpeta

```
usuario@mimaquina: ~  
usuario@mimaquina:~$ ls  
Descargas  Escritorio  Música      Público  
Documentos  Imágenes   Plantillas  Vídeos  
usuario@mimaquina:~$ mv Plantillas Mis_Plantillas  
usuario@mimaquina:~$
```



UD 5 – Gestión del almacenamiento en Linux

rm (remove): borra ficheros o directorios

```
usuario@mimaquina: ~/Imágenes
usuario@mimaquina:~/Imágenes$ ls
leeme.txt  Leeme.txt
usuario@mimaquina:~/Imágenes$ rm leeme.txt Leeme.txt
rm: ¿borrar el fichero regular «leeme.txt» protegido contra escritura? (s/n) s
rm: ¿borrar el fichero regular «Leeme.txt» protegido contra escritura? (s/n) s
usuario@mimaquina:~/Imágenes$
```



No se pueden borrar directorios directamente, hay que usar la opción -r. Entonces se borrarán con todo su contenido

```
usuario@mimaquina: ~
usuario@mimaquina:~$ ls
Descargas  Escritorio  Mis_Plantillas  Público
Documentos  Imágenes    Música           Vídeos
usuario@mimaquina:~$ rm -r Mis_Plantillas
usuario@mimaquina:~$ clear
```

También se pueden usar comodines.

UD 5 – Gestión del almacenamiento en Linux

COMANDOS PARA CAMBIAR PERMISOS Y PROPIETARIO

Las propiedades de los ficheros de Linux nos indican quién es el usuario propietario del fichero y los permisos que tienen los distintos tipos de usuarios del sistema sobre dicho fichero. Observa la siguiente información:

```
kike@kike-VirtualBox:~$ ls -l
total 60
drwxr-xr-x 2 kike kike 4096 sep 23 19:55 Descargas
drwxr-xr-x 2 kike kike 4096 sep 23 19:55 Documentos
drwxr-xr-x 2 kike kike 4096 sep 23 19:55 Escritorio
-rw-r--r-- 1 kike kike 8980 sep 23 19:39 examples.desktop
drwxr-xr-x 2 kike kike 4096 oct  3 16:34 Imágenes
drwxr-xr-x 3 kike kike 4096 sep 24 19:50 'Mi carpeta'
-rw-r--r-- 1 kike kike  5 oct  6 09:02 mifichero.txt
drwxr-xr-x 2 kike kike 4096 sep 23 19:55 Música
drwxr-xr-x 2 kike kike 4096 oct  4 11:34 Nueva
drwxr-xr-x 2 kike kike 4096 sep 23 19:55 Plantillas
drwxr-xr-x 2 kike kike 4096 sep 23 19:55 Público
drwxr-xr-x 4 kike kike 4096 oct  4 20:01 snap
drwxr-xr-x 2 kike kike 4096 sep 23 19:55 Vídeos
kike@kike-VirtualBox:~$
```

El
modificador -
l de ls indica
que quieres
el listado en
formato
LARGO

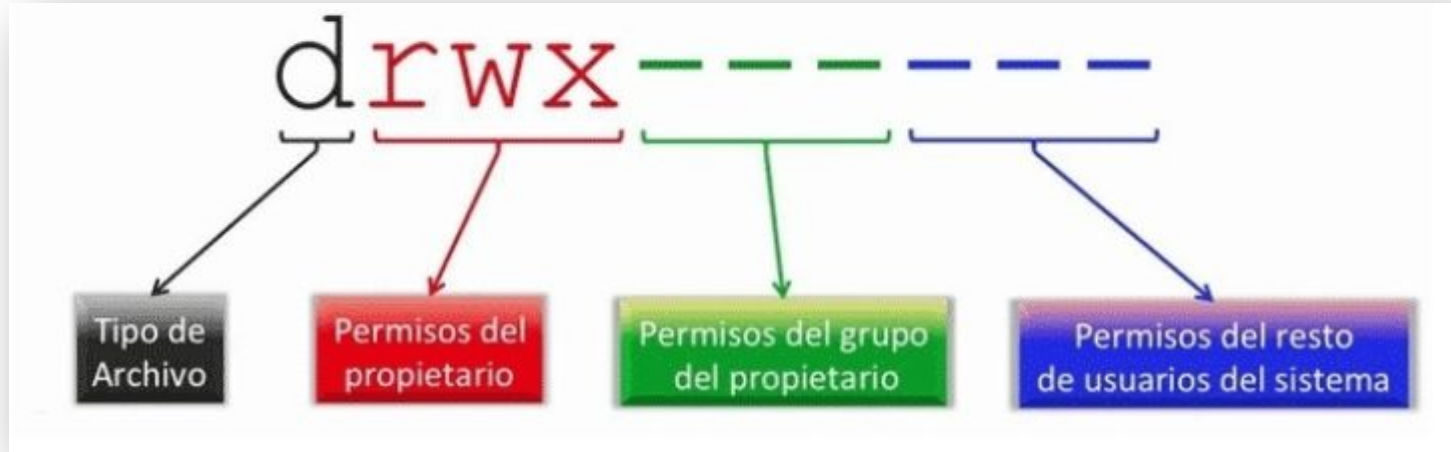
¿Qué
información
identificas?

UD 5 – Gestión del almacenamiento en Linux

Modificador es de permisos	Usuario y Grupo propietario	Tamaño en bytes	Fecha/hora	Nombre del fichero/directorio
<pre>kike@kike-VirtualBox:~\$ ls -l</pre>				
total 60				
drwxr-xr-x	2 kike kike	4096	sep 23 19:55	Descargas
drwxr-xr-x	2 kike kike	4096	sep 23 19:55	Documentos
drwxr-xr-x	2 kike kike	4096	sep 23 19:55	Escritorio
-rw-r--r--	1 kike kike	8980	sep 23 19:39	examples.desktop
drwxr-xr-x	2 kike kike	4096	oct 3 16:34	Imágenes
drwxr-xr-x	3 kike kike	4096	sep 24 19:50	'Mi carpeta'
-rw-r--r--	1 kike kike	5	oct 6 09:02	mifichero.txt
drwxr-xr-x	2 kike kike	4096	sep 23 19:55	Música
drwxr-xr-x	2 kike kike	4096	oct 4 11:34	Nueva
drwxr-xr-x	2 kike kike	4096	sep 23 19:55	Plantillas
drwxr-xr-x	2 kike kike	4096	sep 23 19:55	Público
drwxr-xr-x	4 kike kike	4096	oct 4 20:01	snap
drwxr-xr-x	2 kike kike	4096	sep 23 19:55	Videos
<pre>kike@kike-VirtualBox:~\$</pre>				

UD 5 – Gestión del almacenamiento en Linux

Los modificadores de permisos se codifican del siguiente modo:



d = Directory (directorio)

r = Read (permiso de lectura)

w = Write (permiso de escritura)

x = eXecution (permiso de ejecución para archivos o permiso de acceso/entrada en caso de directorios)

UD 5 – Gestión del almacenamiento en Linux

Ejemplos:

- **rw****r**- - - - -: Archivo con todos los permisos para el propietario, permiso de solo lectura para los usuarios que sean miembros del grupo propietario del archivo. Ningún permiso para el resto de usuarios.

drwx**rw**- **r**- - -: Directorio con todos los permisos para el propietario, permiso de lectura y escritura para los usuarios que sean miembros del grupo propietario del archivo. Solo lectura para el resto de usuarios.

¿Qué hace el sistema cuando un usuario crea un archivo?

```
-rw-r--r-- 1 kike kike 5 oct 6 09:02 mifichero.txt
```

- Establece que el propietario del archivo es el que lo crea (kike)
- El grupo propietario del archivo es el grupo primario del creador (kike)
- Lectura-Escritura para el propietario.
- Solo lectura para los del grupo y el resto de usuarios

UD 5 – Gestión del almacenamiento en Linux

Estos permisos sobre un archivo o carpeta también se suelen denotar como **bloques de tres bits en binario**, de forma que 1 significa ACTIVO y 0 INACTIVO.

Veamos un ejemplo:

rwX **r--** **--X** sería equivalente a: **111 100 000**

Si **representamos en decimal** los números binarios llegamos a esta notación que se utiliza mucho:

111 100 000 sería equivalente a: **740**

Otro ejemplo:

rwX **rw-** **r--** = **111 110 100** = **7 6 4**

Owner	Group	Other
rwx	r-x	r-x
$4+2+1$	$4+0+1$	$4+0+1$
7	5	5

UD 5 – Gestión del almacenamiento en Linux

Rellena la siguiente tabla para practicar:

Notación rwx	Notación binaria	Notación decimal
r-x rw- r--		
	111 101 101	
		650
		651
	111 101 100	
rwX r-X r-X		

UD 5 – Gestión del almacenamiento en Linux

Ahora veamos **los comandos que permiten manipular** tanto los permisos de los archivos como el usuario o grupo propietarios de los mismos.

chmod 754 mifichero: (CHange MODifiers) este comando cambia los modificadores de los permisos del fichero *mifichero*. El usuario que lo ejecuta debe ser el propietario del fichero o hacerlo con permisos de **sudo**, en caso contrario, no funciona.

sudo chown nuevopropietario mifichero: (CHange OWNer) este comando pone a *nuevopropietario* como el nuevo propietario del fichero *mifichero*. Si el usuario que lo ejecuta no lo hace con permisos de **sudo** no funciona.

sudo chgrp nuevogrupo mifichero: (CHange GRouP) este comando cambia el grupo propietario del fichero *mifichero* al grupo *nuevogrupo*. Si el usuario que lo ejecuta no lo hace con permisos de **sudo** no funciona.

