

UD 6. Almacenamiento avanzado en Linux



UD 6. Almacenamiento avanzado en Linux

INDICE DE CONTENIDOS

1.La gestión de dispositivos

2.Tamaño de carpetas y enlaces en Linux

UD 6. Almacenamiento avanzado en Linux

INDICE DE CONTENIDOS

1. La gestión de dispositivos

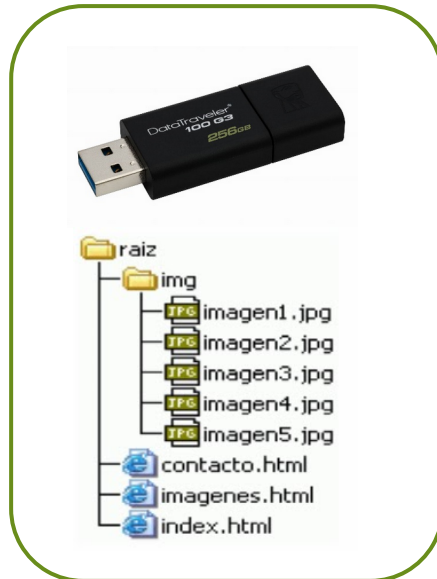
2. Tamaño de carpetas y enlaces en Linux

UD 6. Almacenamiento avanzado en Linux

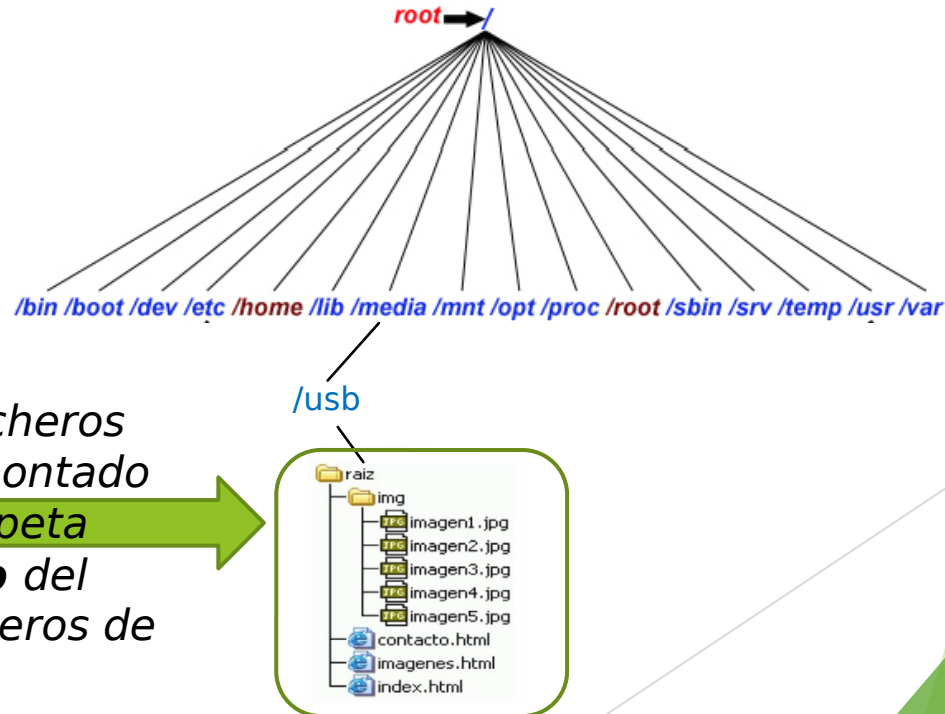
1. La gestión de dispositivos

MONTAJE Y DESMONTAJE DE DISPOSITIVOS

En la unidad anterior vimos que en Linux no existe el concepto de “unidad de disco” (C:, D: ...), como en Windows. En su lugar se usa el concepto de “montar” un sistema de ficheros sobre alguna carpeta del sistema (punto de montaje).

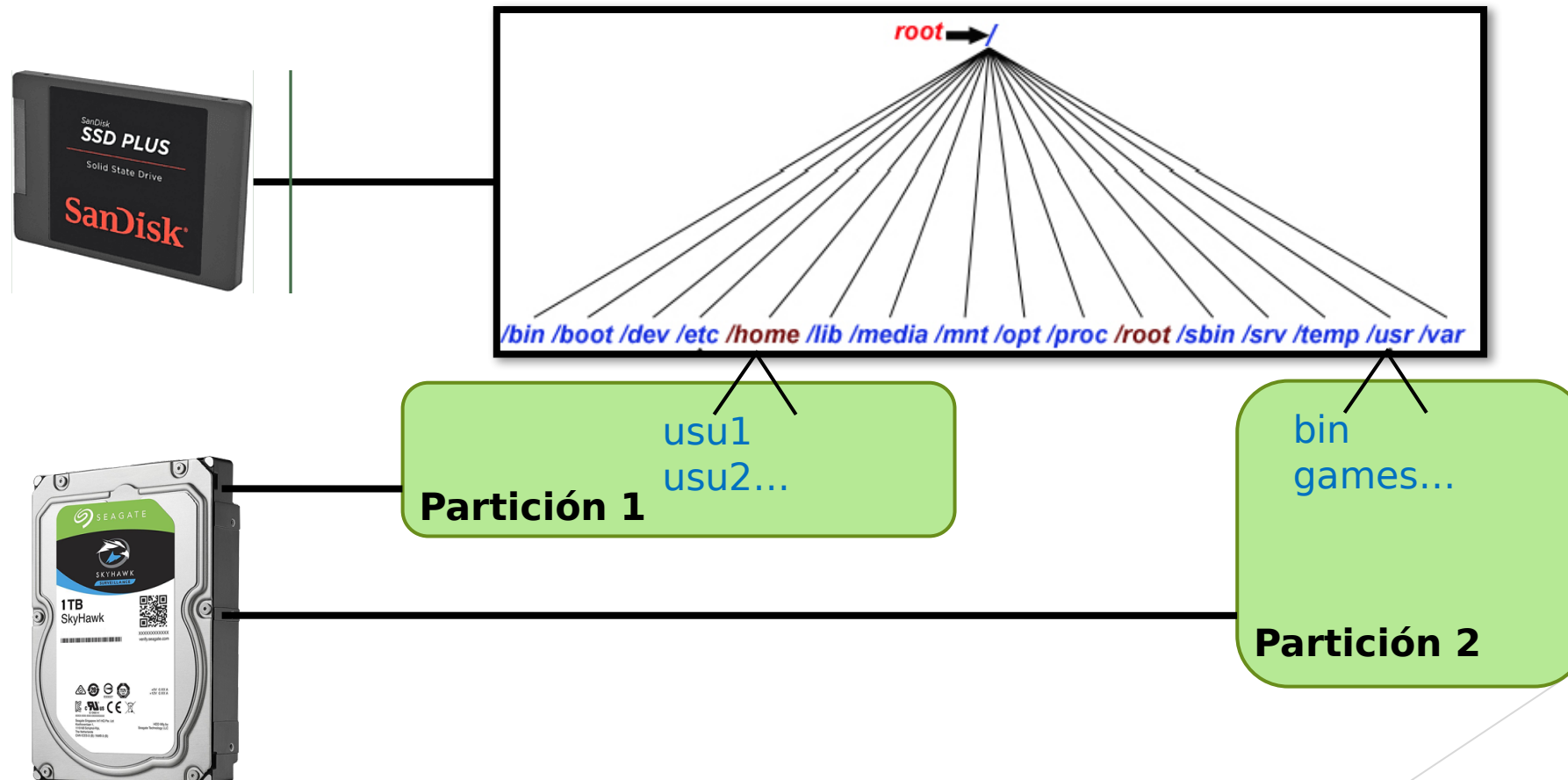


*Sistema de ficheros
del pendrive montado
sobre la carpeta
/media/usb del
sistema de ficheros de
Linux*



UD 6. Almacenamiento avanzado en Linux

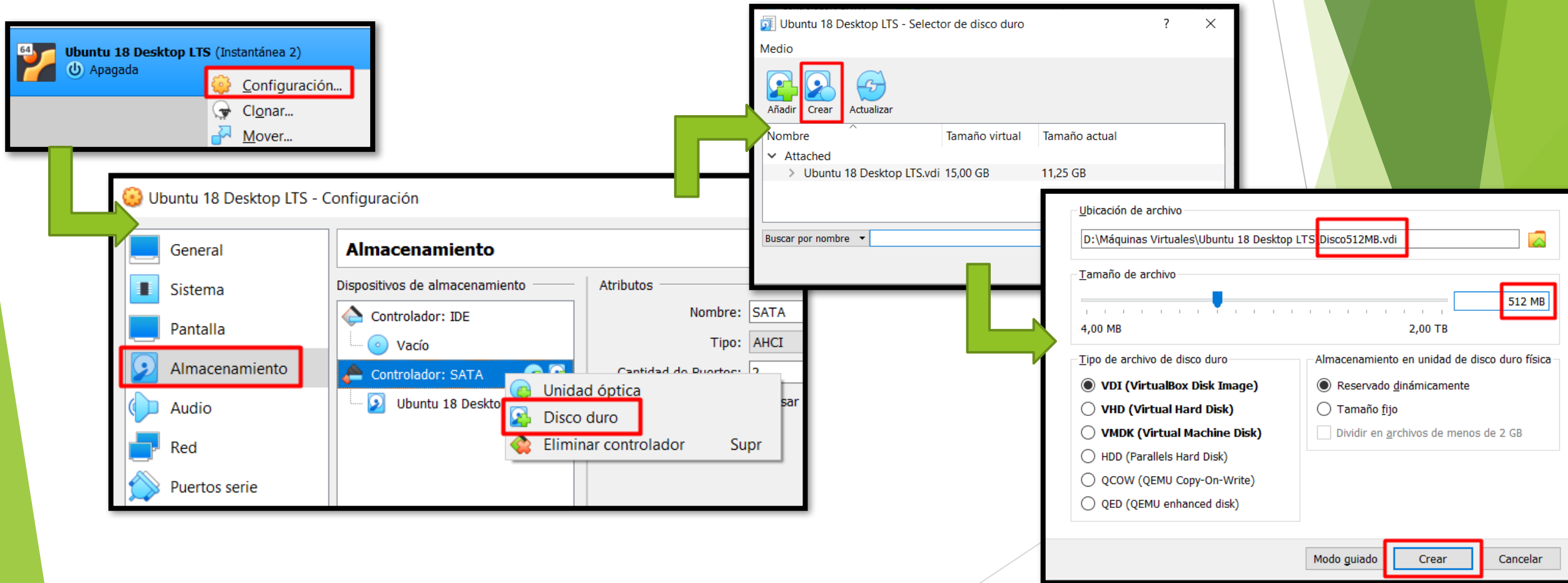
Vimos que este sistema es muy versátil y permite utilizar distintos discos duros y particiones para guardar distintos elementos del sistema:



UD 6. Almacenamiento avanzado en Linux

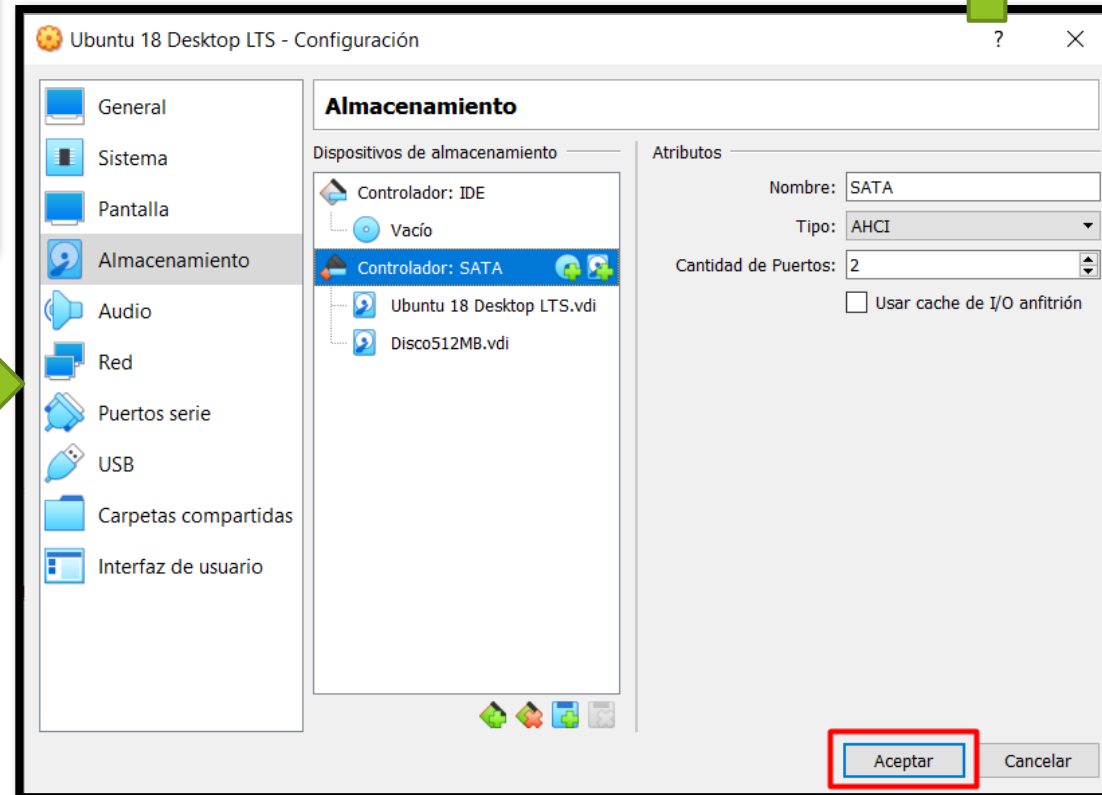
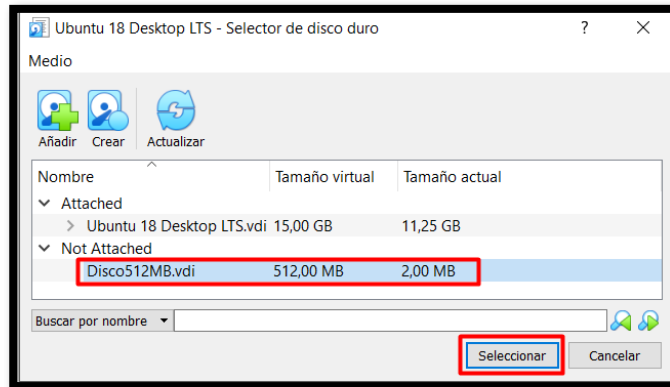
Ahora vamos a ver qué pasos hay que dar para montar una **nueva unidad de disco** en una máquina virtual con Linux:

1) Con la máquina virtual apagada creamos una nueva unidad de disco



UD 6. Almacenamiento avanzado en Linux

2) La seleccionamos, cerramos la configuración y arrancamos la máquina



UD 6. Almacenamiento avanzado en Linux

3) Ahora nuestro dispositivo está “conectado” al sistema pero aún no lo hemos “montado”. Los dispositivos conectados al sistema se pueden ver en la carpeta /dev

```
kike@kike-VirtualBox:~$ cd /dev
kike@kike-VirtualBox:/dev$ ls
autofs      full        loop13      loop9        rtc0         tty10
block       fuse        loop14      loop-control sda          tty11
bsg         hidraw0     loop15      mapper       sda1         tty12
btrfs-control hpet       loop16      mcelog       sdb          tty13
bus         hugepages  loop17      mem          sg0          tty14
cdrom       hwrng      loop18      mqueue       sg1          tty15
char        i2c-0      loop19      net          sg2          tty16
console     initctl    loop2       null         shm          tty17
core        input     loop20      nvram        snapshot     tty18
```

En nuestro caso, nuestro disco duro aparece en /dev con el identificador sdb, ya que es el 2º conectado

Las unidades de discos se codifican con los siguientes nombres:

- Discos sólidos: **/dev/sdXN**
- DVD o CD: **/dev/srXN**
- Disketteras: **/dev/fdXN**

Donde X se sustituye por una letra (a, b, c...) que indica el orden de conexión de los dispositivos (a para el 1º, b para el 2º...)

La N es opcional y, si aparece, representa el número de la partición del dispositivo (1, 2...).

Ejemplo: /dev/sda1 refiere a la primera partición del primer disco sólido conectado

UD 6. Almacenamiento avanzado en Linux

4) Formateamos la unidad a EXT4 con la utilidad mkfs.ext4 (make filesystem)

```
kike@kike-VirtualBox:~$ sudo mkfs.ext4 /dev/sdb
[sudo] contraseña para kike:
mke2fs 1.44.1 (24-Mar-2018)
Se está creando un sistema de ficheros con 131072 bloques de 4k y 32768 nodos-i
UUID del sistema de ficheros: 313a0e4e-bf4d-4922-af4b-bec6655912f4
Respalos del superbloque guardados en los bloques:
    32768, 98304

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (4096 bloques): hecho
Escribiendo superbloques y la información contable del sistema de archivos: hecho
```

5) Montamos la unidad en una carpeta del sistema de archivos de Linux que debe estar creada previamente:

```
kike@kike-VirtualBox:~$ sudo mount /dev/sdb /media/kike
kike@kike-VirtualBox:~$
```



Nuestra unidad ya está montada y lista para utilizarse

6) Cuando queramos dejar de usarla, la podemos “desmontar” con:

```
kike@kike-VirtualBox:~$ sudo umount /dev/sdb
kike@kike-VirtualBox:~$
```

UD 6. Almacenamiento avanzado en Linux

CONSULTAR EL USO DE DISPOSITIVOS

El comando **df** (device free) nos ofrece el espacio libre de cada uno de los dispositivos (reales o virtuales) conectados al equipo, además, de la información de en qué carpeta del sistema de ficheros están “montados”.

Con el modificador **-h** observamos los resultados en formato “humanizado”

/dev/sda1 es el disco duro principal



/dev/sdb es nuestro disco duro secundario

Realizar los ejercicios 1 y 2

```
kike@kike-VirtualBox:~$ df -h
S.ficheros      Tamaño Usados  Disp Uso% Montado en
udev            971M    0      971M  0% /dev
tmpfs           199M    1,6M   198M  1% /run
/dev/sda1       15G    8.9G   5.2G  64% /
tmpfs           994M    0      994M  0% /dev/shm
tmpfs           5,0M    4,0K   5,0M  1% /run/lock
tmpfs           994M    0      994M  0% /sys/fs/cgroup
/dev/loop0      98M     98M    0 100% /snap/core/10577
/dev/loop2     162M   162M    0 100% /snap/gnome-3-28-1804/128
/dev/loop1     218M   218M    0 100% /snap/gnome-3-34-1804/60
/dev/loop3      3,8M   3,8M    0 100% /snap/gnome-system-monitor/100
/dev/loop4     216M   216M    0 100% /snap/wine-platform-5-stable/13
/dev/loop5      98M     98M    0 100% /snap/core/10185
/dev/loop6     384K   384K    0 100% /snap/gnome-characters/570
/dev/loop7      56M     56M    0 100% /snap/core18/1932
/dev/loop8     2,5M   2,5M    0 100% /snap/gnome-calculator/826
/dev/loop9     256M   256M    0 100% /snap/gnome-3-34-1804/36
/dev/loop10     56M     56M    0 100% /snap/core18/1944
/dev/loop11     384K   384K    0 100% /snap/gnome-characters/550
/dev/loop12     1,0M   1,0M    0 100% /snap/gnome-logs/100
/dev/loop14     1,0M   1,0M    0 100% /snap/gnome-logs/61
/dev/loop13     65M     65M    0 100% /snap/gtk-common-themes/1514
/dev/loop15     216M   216M    0 100% /snap/wine-platform-5-stable/12
/dev/loop16     232M   232M    0 100% /snap/wine-platform-runtime/198
/dev/loop17     163M   163M    0 100% /snap/gnome-3-28-1804/145
/dev/loop18     2,3M   2,3M    0 100% /snap/gnome-system-monitor/148
/dev/loop19     2,5M   2,5M    0 100% /snap/gnome-calculator/748
/dev/loop20      63M     63M    0 100% /snap/gtk-common-themes/1506
/dev/loop21     232M   232M    0 100% /snap/wine-platform-runtime/191
tmpfs           199M    28K   199M  1% /run/user/121
tmpfs           199M    36K   199M  1% /run/user/1000
/dev/sdb        488M   780K   452M  1% /media/kike
kike@kike-VirtualBox:~$
```

UD 6. Almacenamiento avanzado en Linux

INDICE DE CONTENIDOS

1.La gestión de dispositivos

2.Tamaño de carpetas y enlaces en Linux

UD 6. Almacenamiento avanzado en Linux

2. Tamaño de carpetas y enlaces en Linux

CONSULTAR EL TAMAÑO USADO DE UNA CARPETA

Con el comando `du` (disk usage) podemos ver el tamaño que ocupa una carpeta y todas sus subcarpetas.

```
kike@kike-VirtualBox:/home$ du kike
4      kike/kk2
4      kike/Escritorio
4      kike/ficheros/apuntes/tema1
8      kike/ficheros/apuntes
12     kike/ficheros
4      kike/Descargas
476    kike/.cache/gnome-software/fwupd/remotes.d/lvfs
480    kike/.cache/gnome-software/fwupd/remotes.d
```

La unidad de medida es el número de bloques de disco que ocupa cada carpeta (normalmente de 1 Kb)

```
4      kike/.dbus
4      kike/.gnupg/private-keys-v1.d
16     kike/.gnupg
4      kike/Apuntes
176960 kike ←
kike@kike-VirtualBox:/home$
```

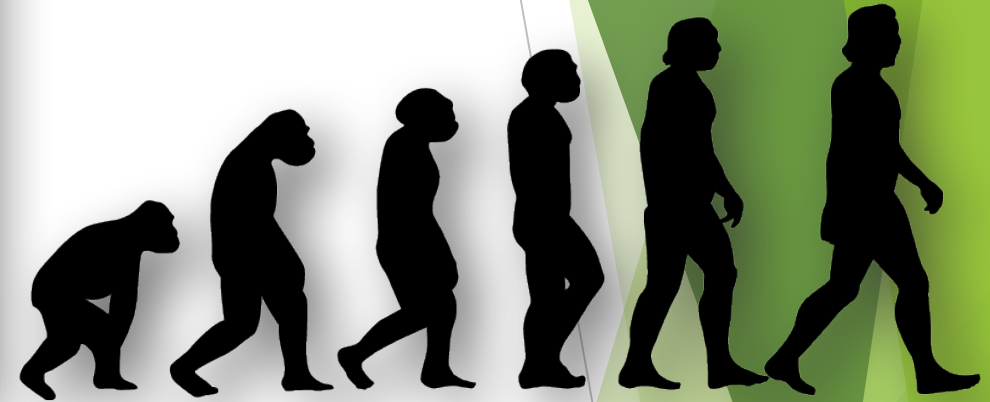
Al final nos muestra el tamaño total de la carpeta kike



UD 6. Almacenamiento avanzado en Linux

Si usamos el modificar `-h` veremos el tamaño en “formato humano”

```
kike@kike-VirtualBox:~$ ls
Anotaciones          Escritorio
Apuntes              examples.desktop
cualquierotronombre.txt ficheros
Descargas            Galletas
Documentos           Imágenes
kike@kike-VirtualBox:~$ du Descargas
4      Descargas
kike@kike-VirtualBox:~$ du -h Descargas
4,0K   Descargas
kike@kike-VirtualBox:~$
```




Si añadimos el modificador `-s` (summary) nos da una sola línea de resumen con el total que ocupa una carpeta sin entrar en el detalle de cada una de sus subcarpetas:

```
kike@kike-VirtualBox:/home$ sudo du -sh kike
174M   kike
kike@kike-VirtualBox:/home$
```

UD 6. Almacenamiento avanzado en Linux

Si no especificamos ninguna carpeta, lo aplica a todos los elementos contenidos en la carpeta actual.

Realizar el ejercicio 1



```
kike@kike-VirtualBox:~$ du -h
4,0K    ./kk2
4,0K    ./Escritorio
4,0K    ./ficheros/apuntes/tema1
8,0K    ./ficheros/apuntes
12K     ./ficheros
4,0K    ./Descargas
476K    ./cache/gnome-software/fwupd/remotes.d/lvfs
480K    ./cache/gnome-software/fwupd/remotes.d
484K    ./cache/gnome-software/fwupd
3,9M    ./cache/gnome-software/icons
```

ENLACES EN LINUX

Los enlaces en Linux tienen un funcionamiento parecido a los “Accesos directos” de Windows y nos permiten acceder a elementos de uso frecuente de nuestro sistema de ficheros tecleando menos.

En Linux existen dos tipos de enlaces:

- Simbólicos o blandos (symbolic / soft link)
- Duros (hard link)

Veamos cada uno de ellos.



UD 6. Almacenamiento avanzado en Linux

Enlaces blandos o simbólicos

Los enlaces simbólicos funcionan como los accesos directos de Windows. Es decir tenemos un “apuntador” a un fichero o una carpeta que nos permite acceder a un recurso dando menos pasos.

Para crearlos tenemos que usar el comando **ln (link)** con la siguiente sintaxis: **ln -s ruta_al_recurso nombre_enlace**

Con **-s** indicamos que el enlace debe ser “simbólico”. Ejemplo:

```
kike@kike-VirtualBox:~/config$ sudo ln -s /etc/netplan/01-network-manager-all.yaml netplan.config
kike@kike-VirtualBox:~/config$ ls -l
total 0
lrwxrwxrwx 1 root root 40 ene  4 12:50 netplan.config -> /etc/netplan/01-network-manager-all.yaml
kike@kike-VirtualBox:~/config$
```

Con **ls -l** vemos el nombre del enlace a dónde apunta

Si ahora hacemos un “**sudo nano netplan.config**” realmente estaremos abriendo y editando el fichero **/etc/netplan/01-network-manager-all.yaml**

Link

UD 6. Almacenamiento avanzado en Linux

También podemos crear enlaces simbólicos a carpetas. Ejemplo:

```
kike@kike-VirtualBox:~$ sudo ln -s /etc etc
kike@kike-VirtualBox:~$ ls -l
total 104
-rw-r--r-- 1 kike kike  0 nov 16 11:57 Anotaciones
drwxr-xr-x 2 kike kike 4096 nov  9 13:02 Apuntes
drwxr-xr-x 2 kike kike 4096 ene  4 12:21 config
-rw-r--r-- 1 kike kike  0 nov  5 08:50 cualquierotronombre.txt
drwxr-xr-x 2 kike kike 4096 nov 16 12:13 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
lrwxrwxrwx 1 root root  4 ene  4 12:31 etc -> /etc
-rwxrwxrwx 1 kike kike 8980 sep 23 19:39 examples.desktop
drwxr-xr-x 3 kike kike 4096 nov  9 13:07 ficheros
```

De este modo podemos tener a mano las carpetas que usemos con más frecuencia:

```
kike@kike-VirtualBox:~$ cd etc
kike@kike-VirtualBox:~/etc$ ls
acpi          dictionaries-common  kerneloops.conf      opt             services
adduser.conf  dpkg                 ldap                  os-release      shadow
alternatives  emacs               ld.so.cache          PackageKit     shadow-
anacrontab    environment         ld.so.conf            pam.conf       shells
apg.conf      firefox             ld.so.conf.d         pam.d           skel
apm           fonts               legal                 papersize       speech-dispatcher
apparmor      fstab               libao.conf            passwd          ssh
apparmor.d    fuse.conf           libaudit.conf        passwd-        ssl
apport        fwupd              libblockdev          pcmcia         subgid
```



UD 6. Almacenamiento avanzado en Linux

El borrado de un enlace simbólico se hace como si se tratara de un fichero normal y corriente:

```
kike@kike-VirtualBox:~/config$ ls
netplan.config
kike@kike-VirtualBox:~/config$ rm netplan.config
kike@kike-VirtualBox:~/config$
```

Y ¿qué pasará con un enlace si borramos el fichero/carpeta al que apunta?

```
kike@kike-VirtualBox:~$ ln -s Documentos/Info-instalación detalles-instala.link
kike@kike-VirtualBox:~$ rm Documentos/Info-instalación
kike@kike-VirtualBox:~$ ls -l
total 68
drwxr-xr-x 2 kike kike 4096 nov  9 13:02 Apuntes
drwxr-xr-x 2 kike kike 4096 ene  4 12:54 config
drwxr-xr-x 2 kike kike 4096 nov 16 12:13 Descargas
lrwxrwxrwx 1 kike kike  28 ene  4 13:00 detalles-instala.link -> Documentos/Info-instalación
drwxr-xr-x 2 kike kike 4096 ene  4 13:00 Documentos
drwxr-xr-x 2 kike kike 4096 sep 23 19:55 Escritorio
```

El enlace se queda “roto” y Linux nos lo marca en rojo para que sepamos que el contenido al que apuntaba no existe.

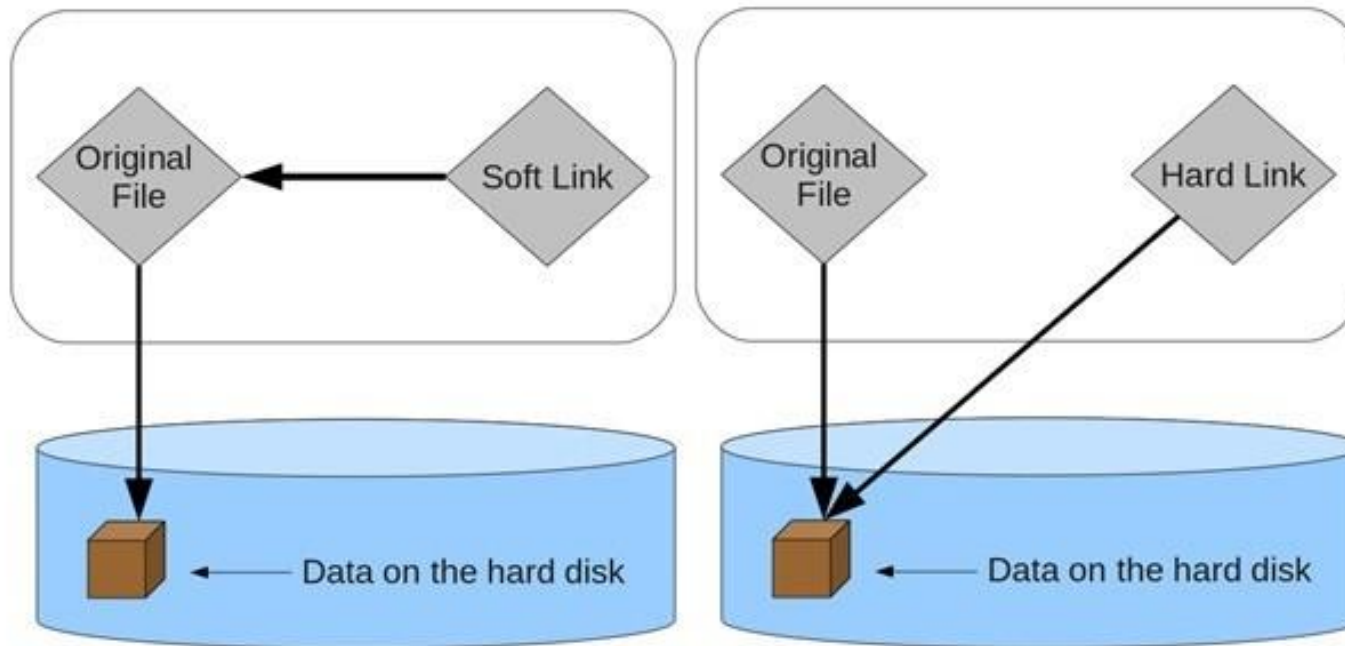


UD 6. Almacenamiento avanzado en Linux

Enlaces duros

Estos enlaces funcionan de una forma distinta y solo pueden ser aplicados a ficheros, nunca a carpetas.

Estos enlaces no apuntan a un fichero, como hacen los enlaces simbólicos, sino **a la zona del disco duro** donde se almacenan sus datos. Fíjate en la diferencia:



HARD
LINKS

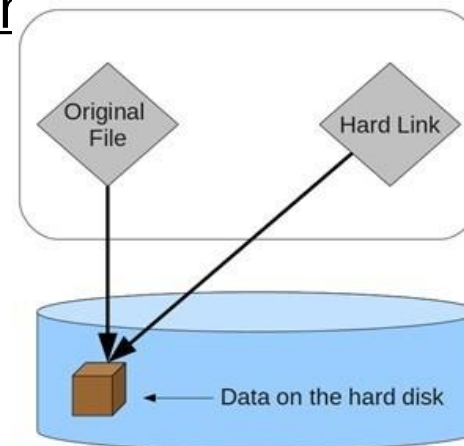
UD 6. Almacenamiento avanzado en Linux

La creación de un enlace duro se realiza igual que la de un enlace simbólico pero sin usar el modificador `-s`. Ejemplo:

```
kike@kike-VirtualBox:~/Descargas$ ls
docum-original.docx
kike@kike-VirtualBox:~/Descargas$ ln docum-original.docx enlace-duro-docum.docx
kike@kike-VirtualBox:~/Descargas$ ls -l
total 520
-rw-rw-r-- 2 kike kike 264579 ene  4 13:37 docum-original.docx
-rw-rw-r-- 2 kike kike 264579 ene  4 13:37 enlace-duro-docum.docx
kike@kike-VirtualBox:~/Descargas$
```

El resultado es que ahora tenemos dos nombres de ficheros (el original y el enlace duro) que apuntan al mismo contenido del disco duro.

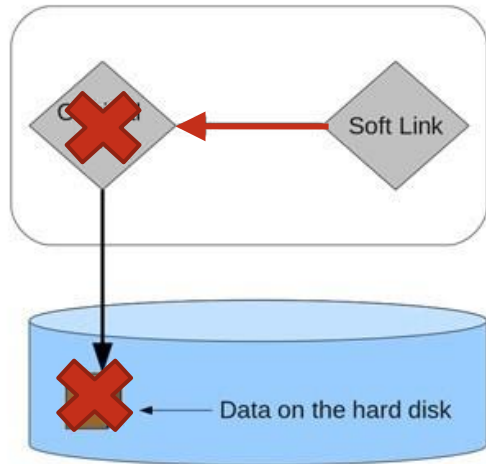
Así que si hacemos los cambios en uno de los dos, los cambios realizados serán visibles en el otro.



UD 6. Almacenamiento avanzado en Linux

En el día a día, los enlaces duros y los simbólicos sirven para lo mismo: “tener que teclear menos”.

La **diferencia** entre ambos se produce **al borrar el fichero original**.

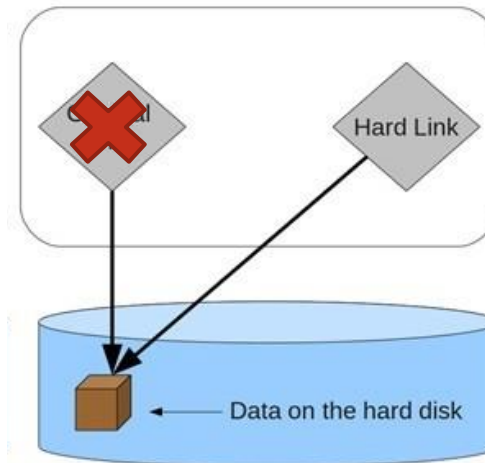


Cuando tenemos un **enlace simbólico** que apunta a un fichero original y borramos este fichero pasan dos cosas:

- Se borran los datos del disco duro
- El enlace simbólico se queda “roto” porque apunta a un fichero que no existe

En cambio, cuando tenemos un **enlace duro** que apunta al contenido del disco duro y borramos el fichero original entonces:

- Solo se borra el fichero original pero no su contenido.
- NO se borran los datos del disco duro porque siguen siendo apuntados por el enlace duro.



Realizar los ejercicios 2 al 4