

Creación y Gestión de Sistemas de Ficheros utilizando Particiones

1. Creamos 4 discos de 1GB de tamaño y los 'conectamos' a nuestra maquina virtual.
2. Una vez hecho esto, comprobamos que se han reconocido los dispositivos con `sudo fdisk -l`

```
(victor@victor)-[~]
$ sudo fdisk -l
[sudo] contraseña para victor:
Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x8502a5b5

Device      Boot      Start        End    Sectors    Size Id Type
/dev/sda1   *          2048    59596799   59594752   28,4G 83 Linux
/dev/sda2                59598846   62912511   3313666    1,6G  f W95 Ext'd (LBA)
/dev/sda5                59598848   62912511   3313664    1,6G 82 Linux swap / Solaris

Disk /dev/sdd: 1 GiB, 1073741824 bytes, 2097152 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdc: 1 GiB, 1073741824 bytes, 2097152 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sde: 1 GiB, 1073741824 bytes, 2097152 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

3. Creamos un sistema de ficheros para cada uno de los discos que hemos conectado con: `sudo mkfs.{FS} /dev/{Disco}`.
- a. `sudo mkfs.ext2 /dev/sdb`
 - b. `sudo mkfs.ext3 /dev/sdc`
 - c. `sudo mkfs.ext4 /dev/sdd`
 - d. `sudo mkfs.xfs /dev/sde`

```
(victor@victor)-[~]
$ sudo mkfs.ext2 /dev/sdb
mke2fs 1.47.2 (1-Jan-2025)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 14642bec-ef99-4cab-8ada-bb197d10fcd6
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

```
(victor@victor)-[~]
$ sudo mkfs.ext3 /dev/sdc
mke2fs 1.47.2 (1-Jan-2025)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: a03110a9-9d28-4111-a134-512a486daedc
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
(victor@victor)-[~]
$ sudo mkfs.ext4 /dev/sdd
mke2fs 1.47.2 (1-Jan-2025)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: fa0df307-c781-4d6d-a849-b6cec77c39e2
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
(victor@victor)-[~]
$ sudo mkfs.xfs /dev/sde
meta-data=/dev/sde          isize=512    agcount=4, agsize=65536 blks
                           =                sectsz=512   attr=2, projid32bit=1
                           =                crc=1        finobt=1, sparse=1, rmapbt=1
                           =                reflink=1     bigtime=1 inobtcount=1 nrext64=1
                           =                exchange=0    metadir=0
data                =                bsize=4096   blocks=262144, imaxpct=25
                           =                sunit=0    swidth=0 blks
naming              =version 2        bsize=4096   ascii-ci=0, ftype=1, parent=0
log                 =internal log     bsize=4096   blocks=16384, version=2
                           =                sectsz=512   sunit=0 blks, lazy-count=1
realtime            =none             extsz=4096   blocks=0, rtextents=0
                           =                rgcount=0    rgsz=0 extents
                           =                zoned=0     start=0 reserved=0
```

4. Creamos el punto de montaje para cada uno de los discos y sus sistema de archivos: `sudo mkdir /mnt/ext2 /mnt/ext3 /mnt/ext4 /mnt/xfs`

```
(victor@victor)-[~]  
$ sudo mkdir /mnt/ext2 /mnt/ext3 /mnt/ext4 /mnt/xfs
```

5. Montamos las particiones en cada uno de los directorios que hemos creado.
- `sudo mount /dev/sdb /mnt/ext2`
 - `sudo mount /dev/sdc /mnt/ext3`
 - `sudo mount /dev/sdd /mnt/ext4`
 - `sudo mount /dev/sde /mnt/xfs`

```
(victor@victor)-[~]  
$ sudo mount /dev/sdb /mnt/ext2
```

```
(victor@victor)-[~]  
$ sudo mount /dev/sdc /mnt/ext3  
(victor@victor)-[~]  
$ sudo mount /dev/sdd /mnt/ext4  
(victor@victor)-[~]  
$ sudo mount /dev/sde /mnt/xfs  
(victor@victor)-[~]  
$
```

6. Verificamos que las particiones se montaron correctamente con: `df -h | grep /mnt`

```
(victor@victor)-[~]  
$ df -h | grep /mnt  
/dev/sdb          1007M    24K   956M    1% /mnt/ext2  
/dev/sdc           975M    60K   924M    1% /mnt/ext3  
/dev/sdd           974M   280K   906M    1% /mnt/ext4  
/dev/sde           960M    51M   910M    6% /mnt/xfs  
(victor@victor)-[~]  
$
```

7. Comparamos las características y rendimiento de los distintos sistemas de archivos creando un archivo de 50MB en cada uno y midiendo el tiempo que tarda en crearse y la tasa de velocidad de escritura.

```
(victor@victor)-[~]
$ sudo dd if=/dev/zero of=/mnt/ext2/testfile bs=1M count=50
50+0 records in
50+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 0,0683636 s, 767 MB/s

(victor@victor)-[~]
$ sudo dd if=/dev/zero of=/mnt/ext3/testfile bs=1M count=50
50+0 records in
50+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 0,0715997 s, 732 MB/s

(victor@victor)-[~]
$ sudo dd if=/dev/zero of=/mnt/ext4/testfile bs=1M count=50
50+0 records in
50+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 0,0375758 s, 1,4 GB/s

(victor@victor)-[~]
$ sudo dd if=/dev/zero of=/mnt/xfs/testfile bs=1M count=50
50+0 records in
50+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 0,0171942 s, 3,0 GB/s

(victor@victor)-[~]
$
```

8. Por ultimo desmontamos y eliminamos los directorios de las particiones para liberar los id y espacio.

```
(victor@victor)-[~]
$ sudo du -sh /mnt/ext2
51M    /mnt/ext2

(victor@victor)-[~]
$ sudo du -sh /mnt/ext3
51M    /mnt/ext3

(victor@victor)-[~]
$ sudo du -sh /mnt/ext4
51M    /mnt/ext4

(victor@victor)-[~]
$ sudo du -sh /mnt/xfs
50M    /mnt/xfs

(victor@victor)-[~]
$
```

```
(victor@victor)-[~]
$ sudo umount /mnt/ext2

(victor@victor)-[~]
$ sudo umount /mnt/ext3

(victor@victor)-[~]
$ sudo umount /mnt/ext4

(victor@victor)-[~]
$ sudo umount /mnt/xfs

(victor@victor)-[~]
$ sudo rmdir /mnt/ext2 /mnt/ext3 /mnt/ext4 mnt/xfs
rmdir: fallo al borrar 'mnt/xfs': No existe el fichero o el directorio
```

Analisis

- **Qué diferencias notaste en la velocidad de escritura entre los sistemas de ficheros?**

El sistema EXT aumenta su velocidad a medida que avanza de version, por ejemplo ext2 presenta 767 MB/s, mientras que ext4 1,4 GB/s. XFS es superior, con una velocidad de 3 GB/s.

- **Cuánto espacio se utilizó realmente para el archivo de prueba de 50 MB en cada sistema de ficheros?**

En todos los sistemas EXT se emplearon 51 MB, es decir, 1 MB extra, mientras que ocupa solamente 50 MB.

- **Por qué hay diferencias?**

Tanto ext4 como xfs estan pensados con filosofías y propósitos diferentes. Ext4 se centra más en la fiabilidad y el uso personal, con características como el journaling, que hace mucho más fáciles las recuperaciones, y la capacidad de agrandar y reducir volúmenes, muy útil para *home users*, además funciona de manera muy eficiente con archivos pequeños y es muy estable, aunque con el tiempo se observan problemas de fragmentación y una menor eficiencia con archivos muy grandes.

Xfs por su parte esta pensado para trabajar con grandes volúmenes de archivos y archivos de gran tamaño, además de poseer una fragmentación mínima. Sin embargo cuenta con la principal desventaja de no ser tan eficiente como ext4 con archivos pequeños, y sobre todo la de no poder reducir su volumen, solo agrandarlo.

- **Qué tipo de aplicaciones crees que se beneficiarían más del uso de XFS?**

Servidores que manejen grandes bases de datos, donde pueden existir muchos archivos y muy pesados, y también en sistemas de almacenamiento, donde no se planea una reducción del espacio.

- **Por qué sería importante elegir el sistema de ficheros adecuado según el uso previsto?**

Es crucial para ayudar con la gestión eficiente de un sistema y sus aplicaciones. Emplear ext4 en una base de datos puede suponer una mayor demora en las consultas y sentencias de modificación a causa del journaling. Del mismo modo un ordenador personal no se vería beneficiado por las bondades de xfs, pero si sufriría sus carencias, como la falta de recuperación ante errores por falta de transaccionalidad.