

## 0.

### Objetivos del aprendizaje

- Explicar la importancia de la administración del sistema de ficheros en la labor de administración de sistemas informáticos.
- Explicar que sucede en el sistema operativo cuando se montan y desmontan dispositivos en el sistema de ficheros.
- Utilizar la herramienta `mount` (y `umount`) para montar (y desmontar) sistemas de ficheros.
- Configurar los dispositivos de almacenamiento principales del sistema mediante el fichero `/etc/fstab`.
- Explicar la necesidad del chequeo del sistema de ficheros y utilizar las principales herramientas de chequeo.
- Explicar cómo funciona el sistema de *journaling*.
- Enumerar los pasos necesarios para añadir un nuevo dispositivo de disco al sistema operativo.
- Explicar las diferencias fundamentales entre los sistemas de ficheros `ext2`, `ext3` y `ext4`.
- Justificar la necesidad del uso de cuotas de disco en sistemas informáticos compartidos.
- Configurar cuotas de disco para los distintos usuarios de un sistema GNU/Linux.
- Justificar la necesidad del uso de arrays redundantes de discos (RAID) en sistemas informáticos.
- Elegir el mecanismo RAID más adecuado: RAID0, RAID1 o RAID4/5.
- Ejemplificar el mecanismo de redundancia para RAID basado en paridad.
- Montar un sistema RAID por *software* en GNU/Linux.

### Contenidos

#### 7.1. Introducción

#### 7.2. Montaje y desmontaje de sistemas de ficheros.

##### 7.2.1. Concepto.

##### 7.2.2. Herramienta `mount`.

##### 7.2.3. Fichero `/etc/fstab`.

#### 7.3. Comprobación del sistema de ficheros.

- 7.3.1. Concepto y herramientas de chequeo.
- 7.3.2. Sistema de *journaling* para sistemas de ficheros.
- 7.4. Creación de sistemas de ficheros.
  - 7.4.1. Pasos para la creación de un sistema de ficheros.
  - 7.4.2. Diferencias entre `ext2`, `ext3` y `ext4`.
- 7.5. Aspectos avanzados.
  - 7.5.1. Cuotas.
    - 7.5.1.1. Concepto de cuotas de disco.
    - 7.5.1.2. Pasos necesarios para establecer cuotas de disco.
  - 7.5.2. Administración de volúmenes dinámicos (RAID).
    - 7.5.2.1. RAID nivel 0, nivel 1 y nivel 4/5.
    - 7.5.2.2. Uso de la paridad para redundancia de datos.
    - 7.5.2.3. Pasos necesarios para montar un sistema RAID.

## Evaluación

- Cuestionarios objetivos.
- Pruebas de respuesta libre.
- Tareas de administración.

# 1. Introducción

## Introducción

*Tareas asociadas al sistema de ficheros en GNU/Linux:*

- Permitir el acceso a ficheros locales y remotos.
- Controlar los recursos que proporcionan: cuotas de disco, bloques o nodos-i libres, estadísticas de uso...
- Proteger de posibles fallos o errores: RAID, copias de seguridad...
- Controlar y proporcionar seguridad de los datos: propietarios, permisos...
- Durante el arranque: chequear (y corregir) el sistema de ficheros.
- Instalar y configurar nuevos dispositivos de almacenamiento.

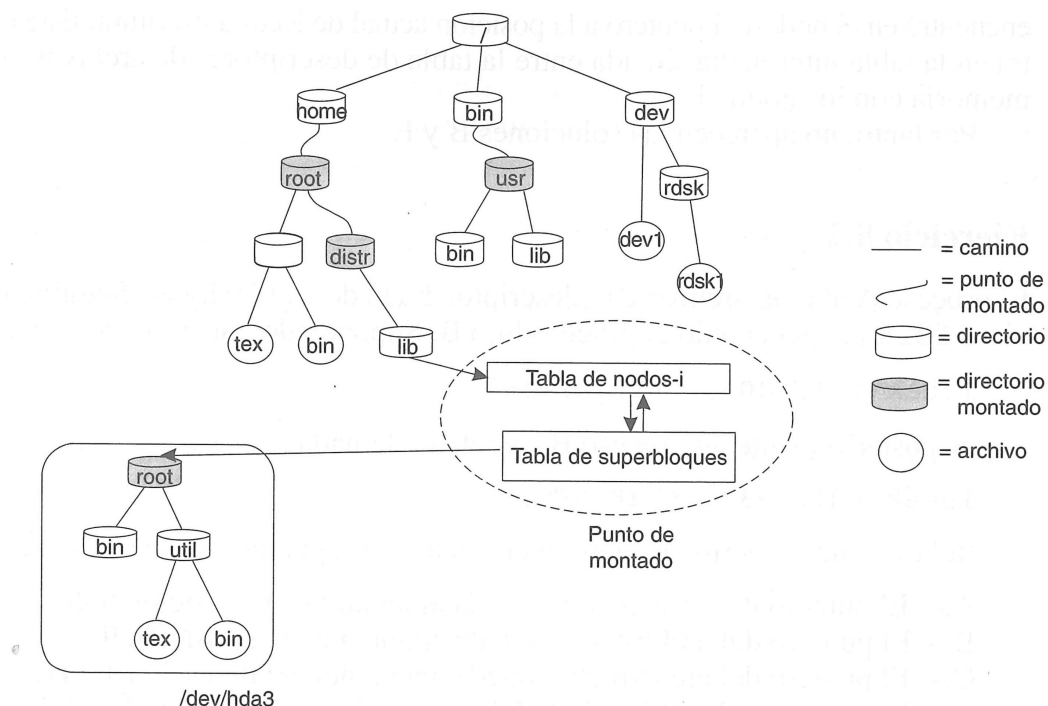
## 2. Montaje y desmontaje de sistemas de ficheros

### 2.1. Concepto

#### Montar y desmontar sistemas de ficheros

- En GNU/Linux, hay un único sistema de ficheros lógico (o una única “jerarquía de directorios”), en la que se organizan todos los dispositivos de almacenamiento disponibles.
- Cada partición tiene su propio sistema de ficheros, con su directorio raíz y su jerarquía.
  - *Montar un sistema de ficheros*: añadirlo al sistema de ficheros lógico. Sus datos (desde su propia raíz) están disponibles a partir de un punto de montaje (p.ej. /media/usb1).
  - *Desmontar un sistema de ficheros*: el sistema de ficheros deja de estar disponible, dejándolo además consistente.
- Los ficheros principales del SO están siempre disponibles desde la raíz del sistema de ficheros lógico (/).
- En el arranque, se monta primero la partición correspondiente a dicha raíz (root) y luego cualquier partición auxiliar.

#### Montar y desmontar sistemas de ficheros



## 2.2. Herramienta mount

### Montar y desmontar sistemas de ficheros

- `mount [opci] <FicheroEspecialBloque> <PtoMontaje>`
  - `-t tipo-sf` ⇒ tipo de sistema de ficheros.
  - `-r` ⇒ montaje en modo sólo lectura.
  - `-w` ⇒ montaje en modo lectura/escritura.
  - `-o opcionesMontaje` ⇒ opciones del proceso de montaje (`nosuid`, `exec`, `remount`, etc.).
- `umount <PtoMontaje>` (ó `<FicheroEspecialBloque>`) ⇒ desmontar un sistema de ficheros. Si está siendo utilizado (*busy*), no se podrá desmontar.
- `fuser` ⇒ saber qué ficheros se están usando y qué procesos los usan (f: fichero abierto, c: directorio de trabajo, e: ejecutando un fichero, etc.)
- `lsof` ⇒ obtener un listado de todos los ficheros abiertos...

### Montar y desmontar sistemas de ficheros

```

1 pedroa@pagutierrezLaptop:~$ fuser -mv / # -m: ficheros montados; -v: verbose
2                                USER      PID ACCESS COMMAND
3 /:                             root      kernel mount /
4                                pedroa    2363 Frce. gnome-keyring-d
5                                pedroa    2760 Fr.e. icedove-bin
6                                pedroa    3206 Fr.e. evince
7 pedroa@pagutierrezLaptop:~$ lsof
8 COMMAND  PID  USER   FD TYPE DEVICE SIZE/OFF      NODE NAME
9 ...
10 kile     2764 pedroa  mem REG   8,4    499320 5246682 libgcrypt.so.11.6.0
11 kile     2764 pedroa  mem REG   8,4    659656 5775741 libgnutls.so.26.14.12
12 kile     2764 pedroa  mem REG   8,4    68416 5775700 libavahi-client.so.3.2.9
13 kile     2764 pedroa  mem REG   8,4    47448 5775702 libavahi-common.so.3.5.3
14 kile     2764 pedroa  mem REG   8,4    14480 5775706 libavahi-glib.so.1.0.2
15 kile     2764 pedroa  mem REG   8,4     4016 5770905 libcanberra.so.0.2.5
16 kile     2764 pedroa  mem REG   8,4    31304 5771046 libgailutil.so.18.0.1
17 ...
18 lsof     3271 pedroa  cwd DIR   8,4      4096 6301896 /home/pagutierrez
19 lsof     3271 pedroa  rtd DIR   8,4      4096 2 /
20 lsof     3271 pedroa  txt REG   8,4   131312 5767875 lsof
21 lsof     3271 pedroa  mem REG   8,4    68352 5774521 locale-archive
22 lsof     3271 pedroa  mem REG   8,4   642216 5248383 libc-2.13.so
23 lsof     3271 pedroa  mem REG   8,4   141088 5242884 ld-2.13.so

```

## 2.3. Fichero /etc/fstab

### Montar y desmontar sistemas de ficheros

- `/etc/fstab`: fichero con información sobre todos los sistemas de ficheros a montar (o ya montados) y las zonas de intercambio a activar.

```
fi_especial pto tipo opciones dump_freq pass_num
```

- `file especial` ⇒ fichero especial de bloques (`/dev/...`).
- `pto` ⇒ directorio que sirve de punto de montaje (*¿permisos?*).
- `tipo` ⇒ tipo de SF (`ext2, ext3, ext4, vfat, iso9660, swap, ntfs, nfs, etc.`).
- Opciones para el proceso de montaje (separadas por “,” y sin espacios).
- `dump_freq` ⇒ “frecuencia del *dump*” para hacer una copia de seguridad de ese SF mediante el comando `dump` (no se usa).
- `pass_num` ⇒ en tiempo de arranque, en qué orden hay que chequear los SFs (ejecutar `fsck` para comprobar su estado).

### Opciones de `fstab`

#### ■ Opciones del fichero `/etc/fstab`:

- `rw` ⇒ Lectura-escritura (por defecto).
- `ro` ⇒ Sólo lectura
- `suid/nosuid` ⇒ Permitido (o no) que los bits `suid` o `sgid` tengan efecto.
- `auto/noauto` ⇒ Montar automáticamente (o no) (ejecutando `mount -a` se montan todos los automáticos ⇒ siempre se ejecuta al arrancar el sistema).
- `exec/noexec` ⇒ Permitir (o no) la ejecución de ficheros.
- `usrquota, grpquota` ⇒ Activar cuotas.
- `uid=500, gid=100` ⇒ Propietario y grupo propietario de los ficheros del SF (si el SF no incorpora esta información o si se quiere cambiar).
- `umask=137` ⇒ Permisos de los ficheros (en este caso, 640) (si el SF no incorpora esta información o si se quieren cambiar).
- `dev` ⇒ Interpretar ficheros especiales en el sistema de archivos.
- `sync` ⇒ Forzar a que todas las operaciones sean síncronas (puede disminuir el tiempo de vida de la unidad de disco).
- `user` ⇒ permite que los usuarios puedan montar el sistema de ficheros. Solo el mismo usuario podrá desmontarlo. Implica las opciones `noexec, nosuid y nodev`.
- `users` ⇒ igual que `user` pero cualquiera podrá desmontarlo.
- `nouser` ⇒ Solo `root` puede montar el SF.
- `owner` ⇒ permite que un usuario pueda montar el sistema de ficheros, siempre que sea dueño del fichero de dispositivo. Implica las opciones `nosuid y nodev`.
- `defaults` ⇒ `rw, suid, dev, exec, auto, nouser, async`.

### Montar y desmontar sistemas de ficheros

#### ■ Ejemplo de contenido del fichero `/etc/fstab`:

1	<code>LABEL=/</code>	<code>/</code>	<code>ext3</code>	<code>defaults,usrquota</code>	<code>0 1</code>
2	<code>/dev/sda3</code>	<code>/windows</code>	<code>vfat</code>	<code>defaults</code>	<code>0 0</code>
3	<code>/dev/dvd</code>	<code>/media/dvd</code>	<code>iso9660</code>	<code>noauto,owner,ro</code>	<code>0 0</code>
4	<code>/dev/fd0</code>	<code>/media/floppy</code>	<code>vfat</code>	<code>noauto,uid=500</code>	<code>0 0</code>
5	<code>/dev/sda4</code>	<code>/otrolinux</code>	<code>ext3</code>	<code>defaults</code>	<code>0 2</code>
6	<code>/dev/sda2</code>	<code>swap</code>	<code>swap</code>	<code>defaults</code>	<code>0 0</code>

- Al ejecutar `mount` como *root*:
  - `mount /media/dvd`: coge las opciones que faltan del fichero.
  - `mount -t iso9660 -r /dev/dvd /media/dvd`: no las coge.
- Si se asigna permisos de montaje a los usuarios (opciones `user`, `users` u `owner`), sólo pueden ejecutar `mount /media/dvd` (sin opciones).
- `mount -a`: montar todas las unidades que sean `auto`.
- Automontado de unidades: `udev` y `dbus`.

### 3. Comprobación del sistema de ficheros

#### 3.1. Concepto y herramientas de chequeo

##### Comprobación del sistema de ficheros

- Durante el arranque, `fsck` o `e2fsck` chequearán la consistencia o estado del sistema de ficheros, detectando problemas e intentando repararlos.
- Se actúa sobre la estructura (no sobre el contenido):
  - Bloques que pertenezcan a varios ficheros.
  - Bloques que están marcados como libres, pero que se encuentran en uso.
  - Bloques que se encuentran marcados como en uso, pero que están libres.
  - Inconsistencias en cuanto al número de enlaces hacia un nodo-i.
  - Nodos-i marcados como libres, pero que están en uso.
  - Nodos-i marcados como en uso, pero que están libres.

##### Comprobación del sistema de ficheros

- Para chequear un SF siempre debe estar desmontado o montado en modo de sólo lectura.
- El SF raíz debe estar montado en modo de sólo lectura (el SF raíz no se puede desmontar, ¿por qué?).
- Si al arrancar el proceso de chequeo encuentra problemas que no puede solucionar, obliga al administrador a que realice el chequeo “a mano” ejecutando la orden `fsck` o `e2fsck` (modo monousuario).

### 3.2. Sistema de journaling

#### Comprobación del sistema de ficheros

- *Journaling*: para evitar la verificación completa (con `fsck`) de sistemas de ficheros de gran tamaño, que sería muy costosa, se implementa un modelo de control transaccional basado en *logging* (un diario).
  - Las suboperaciones que modifiquen los metadatos y datos de un mismo archivo se agrupa en la misma transacción.
  - Si el sistema falla, las acciones parcialmente realizadas se deshacen o completan, recorriendo el *log*.
  - No se garantiza que el sistema esté actualizado al finalizar la recuperación, sino que es consistente.
- Sistemas con esta filosofía: JFS (IBM), `ext3` y la gran mayoría de sistemas de archivos modernos.

#### Comprobación del sistema de ficheros

```
1 TxBegin(&tid);  
2 /* ...  
3   Sub-operaciones  
4   ... */  
5 TxCommit(tid);  
6 TxEnd(tid);
```

- Por cada sub-operación que altera las estructuras de disco se escribe un registro en el *log*, que incluye las modificaciones en los *buffers* de i-nodos y de bloques.
- Cuando se ha copiado a disco (*log*) el registro de *commit*, se empiezan a procesar realmente los *buffers*.
- Después de una caída:
  - Se completan las transacciones *committed*.
  - Se descartan el resto de transacciones.

## 4. Creación de sistemas de ficheros

### 4.1. Pasos necesarios

#### Creación del sistema de ficheros

##### Añadir un nuevo disco o SF:

1. Realizar la conexión física.
2. Crear un fichero especial de dispositivo (si es necesario).
3. Crear las particiones: `fdisk` (o `parted`).

4. Crear sistema de ficheros: `mke2fs -t ext2 /dev/sdb3`
5. Etiquetar la partición usando `e2label`  $\Rightarrow$  asigna una etiqueta al SF que se puede usar en el fichero `/etc/fstab`, en el campo `fi.especial`, mediante `LABEL=etiqueta`.
6. Crear el directorio que hará de punto de montaje.
7. Montar el nuevo sistema de ficheros.
8. Actualizar `/etc/fstab` con las opciones necesarias.

## 4.2. Diferencias entre **ext2**, **ext3** y **ext4**

### Creación del sistema de ficheros

- Diferencias `ext2`, `ext3` y `ext4`:
  - `ext3` tiene el mismo formato que `ext2` pero además es transaccional: añade un registro o *journal* que permite recuperar la consistencia tras una caída del sistema.
  - `ext4` tiene un formato similar a `ext3`, pero además incluye:
    - Una extensión describe un conjunto de bloques lógicos contiguos de un fichero que también se encuentran contiguos en disco: muy útil para ficheros grandes<sup>1</sup>.
    - Se retrasa la reserva de bloques de disco hasta que se va a escribir en él: mayor número de bloques contiguos en disco.
    - Implementa una herramienta de desfragmentación *online*, `e4defrag` (capaz de funcionar mientras se usa el SF).
    - Manejo de sistemas de ficheros y ficheros de mayor tamaño.

### Creación del sistema de ficheros

- ¿Qué sistema elegir?
  - `ext2`  $\Rightarrow$  muy rápido en general, pero no tiene *journaling*. Se puede usar en un SF en el que se guardarán ficheros temporales.
  - `ext3`  $\Rightarrow$  buen rendimiento en general y *journaling*.
  - `ext4`  $\Rightarrow$  menor uso del CPU y mayor rapidez en los procesos de lectura y escritura que `ext3`. Estándar de facto en Linux.

### Creación del sistema de ficheros

- `tune2fs`  $\Rightarrow$  Conocer y ajustar parámetros de un SF `ext4/ext3/ext2`.
  - `-l dispositivo`: Listar el contenido del superbloque del SF.
  - `-c max-mount-counts dispositivo`: Establecer el nº de montajes máximo sin realizar un `fsck`.

---

<sup>1</sup>[https://ext4.wiki.kernel.org/index.php/Ext4\\_Design](https://ext4.wiki.kernel.org/index.php/Ext4_Design)



- `-i numero[d|m|w] dispositivo`: Indicar el tiempo máximo entre dos chequeos.
- `-L etiqueta dispositivo`: Poner una etiqueta al sistema de ficheros.
- `-m porcentaje dispositivo`: Fijar el porcentaje de bloques reservados para procesos especiales (de root). Por defecto, 5 %.

### Parámetros del SF ext4

```

1 tune2fs 1.43.4 (31-Jan-2017)
2 Filesystem volume name:      <none>
3 Last mounted on:            /home
4 Filesystem UUID:            1d8472c1-ea34-4384-a183-367d60e6842f
5 Filesystem magic number:    0xEF53
6 Filesystem revision #:      1 (dynamic)
7 Filesystem features:        has_journal ext_attr resize_inode dir_index filetype
                             needs_recovery extent flex_bg sparse_super large_file huge_file uninit_bg dir_nlink
                             extra_isize
8 Filesystem flags:           signed_directory_hash
9 Default mount options:     user_xattr acl
10 Filesystem state:          clean
11 Errors behavior:           Continue
12 Filesystem OS type:        Linux
13 Inode count:               10993664
14 Block count:               43959808
15 Reserved block count:     2197990
16 Free blocks:               16795196
17 Free inodes:               10106426
18 First block:               0
19 Block size:                4096
20 Fragment size:             4096
21 Reserved GDT blocks:      1013
22 Blocks per group:          32768
23 Fragments per group:       32768
24 Inodes per group:          8192
25 Inode blocks per group:    512
26 Flex block group size:     16
27 Filesystem created:        Tue Oct 11 12:57:37 2016
28 Last mount time:           Wed Apr 3 09:01:25 2019
29 Last write time:           Wed Apr 3 09:01:25 2019
30 Mount count:               535
31 Maximum mount count:       -1
32 Last checked:              Tue Oct 11 12:57:37 2016
33 Check interval:            0 (<none>)
34 Lifetime writes:           1666 GB
35 Reserved blocks uid:       0 (user root)
36 Reserved blocks gid:       0 (group root)
37 First inode:               11
38 Inode size:                 256
39 Required extra isize:       28
40 Desired extra isize:        28
41 Journal inode:              8
42 First orphan inode:         4469691
43 Default directory hash:     half_md4
44 Directory Hash Seed:        77bb1657-5204-456a-87f2-cc27e6b7830b
45 Journal backup:             inode blocks

```

## 5. Aspectos avanzados

### 5.1. Cuotas

#### Concepto de cuotas de disco

##### Cuotas de disco

Permiten limitar el número de bloques y/o ficheros (nodos-i) que un usuario puede usar en una partición (también se pueden establecer para grupos de usuarios).

- Hay dos tipos de límites:
  - Límite *hard*: el usuario no puede sobrepasarlo. Si lo hace, ya no podrá usar más bloques o crear más ficheros.
  - Límite *soft*: es inferior al límite *hard* y se puede sobrepasar durante cierto tiempo, siempre que no se alcance el límite *hard*.
- *Periodo de gracia*: tiempo durante el que se puede sobrepasar el límite *soft*. Se informa al usuario de que ha superado el límite y que debe liberar espacio o nodos-i (ficheros).
  - Los periodos y los límites se establecen, de forma independiente, para bloques y nodos-i.

#### Establecer cuotas de disco

- Pasos a realizar para establecer las cuotas de disco:

1. Instalar el paquete `quota`:

```
1 apt-get install quota
```

2. Indicarlo en `fstab` (diferente en `ext3` y `ext4`):

```
1 /dev/sdb1 /home ext4 defaults,usrjquota=aquota.user,grpjquota=aquota.group,  
jqfmt=vfsv0 0 2
```

```
1 /dev/sdb1 /home ext3 defaults,usrquota,grpquota 0 2
```

3. Remontar la partición para que se activen las opciones: `mount -o remount /home`.
4. `quotacheck -avugm`: añade el contenido de los ficheros de control de cuotas.
  - a: todos los dispositivos con cuotas.
  - v: *verbose*.
  - u: cuotas para usuarios.
  - g: cuotas para grupos.
  - m: no remontar los archivos en modo solo lectura.
5. Activar las cuotas: `quotaon -avug`.
6. Desactivarlas: `quotaoff -avug`.
7. Editar la cuota del usuario `pagutierrez`: `edquota pagutierrez`

```

1 Cuotas de disco para user pagutierrez (uid 1008):
2   Sist. arch. bloques    blando    duro inodos blando    duro
3   /dev/sdb1  39884712 130000000 140000000 507523      0      0

```

## 8. Establecer el periodo de gracia: `edquota -t`

```

1 Período de gracia antes de imponer límites blandos para users:
2 La unidad de tiempo puede ser: días, horas, minutos, o segundos
3   Sist. arch.          Periodo gracia bloque   Periodo gracia inodo
4   /dev/sdb1              7día                  7día

```

## 9. Copiar cuotas: `edquota -up pagutierrez jsanchez`

## 10. Estadísticas de las cuotas: `repquota /dev/sdb1`

```

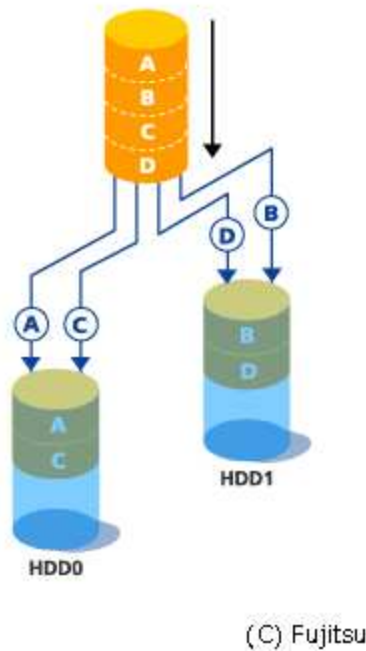
1 *** Informe para user quotas en dispositivo /dev/sdb1
2 Periodo de gracia de bloque: 7días; periodo de gracia de inodo: 7días
3                               Límites de bloque límites de archivo
4 Usuario          usado      blando      duro      usado blando duro
5 -----
6 root             --  10093264      0          0      199316  0    0
7 pagutierrez --  39884712 130000000 140000000  507523  0    0
8 i22fenaf --    31940744 130000000 140000000  864578  0    0
9 jsanchezm --    31940744 130000000 140000000  864578  0    0

```

## 5.2. Administración de volúmenes dinámicos

### Administración de volúmenes dinámicos

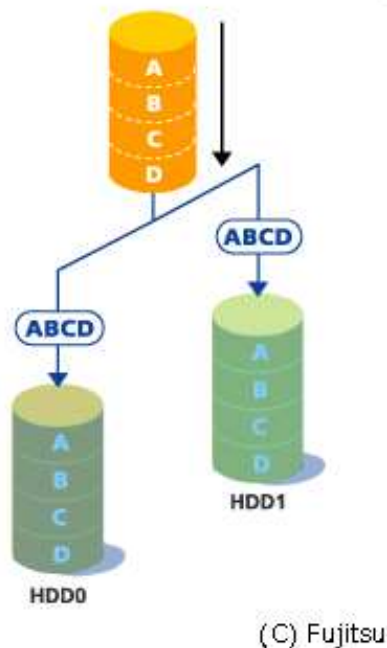
- **RAID:** Array redundante de discos independientes.
  - Varias unidades de disco se ven como una sola unidad lógica.
  - Se pueden implementar por *software* o por *hardware*.
- **Logical Volume Management (LVM):** agrupar las particiones en volúmenes. Permite re-dimensionar y mover particiones.
- **RAID nivel 0:**
  - Expande la información en diversos discos, que se ven como un único SF.
  - Aumenta el espacio según el número de discos usado.
  - Se consigue E/S paralela en lecturas y escrituras, siempre que los bloques a tratar no sean del mismo disco.
  - No hay redundancia de datos.



### Administración de volúmenes dinámicos

#### ■ RAID nivel 1:

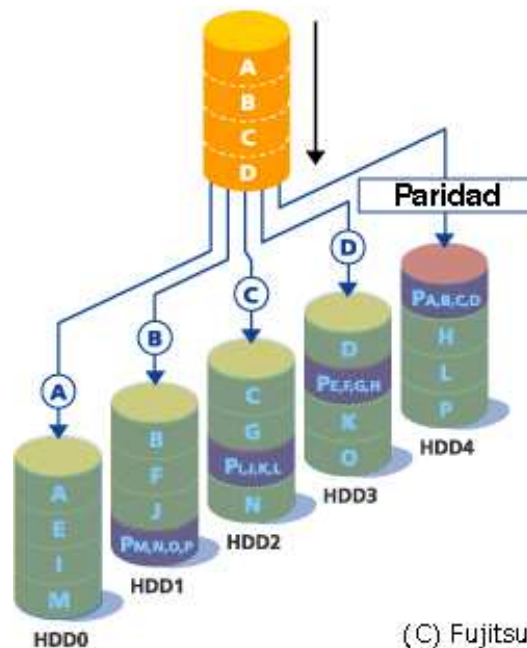
- Se utilizan dos o más discos duros, que forman un único SF (SF replicado en varios discos).
- Son discos espejos (todos guardan la misma información).
- *SI* hay redundancia de datos.
- Las lecturas pueden ser en paralelo, las escrituras no.
- Cuando uno de los discos falla, el sistema sigue trabajando con el otro sin problemas.
- La recuperación de un disco es transparente al usuario.



### Administración de volúmenes dinámicos

- RAID nivel 4/5:
  - División de los datos a nivel de bloques.
  - RAID 4: Mínimo 3 discos duros, de los cuales 1 almacenará la paridad de los otros discos, que son usados para datos.
  - Problema: el disco con paridad es un cuello de botella. RAID 5: repartir paridad entre todos los discos.
  - Se consigue un dispositivo de almacenamiento más grande.
  - *SI* hay redundancia de datos.
  - Lecturas y escrituras en *paralelo*.
  - RAID 5 ofrece la mejor relación rendimiento-coste en un entorno con varias unidades.

RAID5



### Administración de volúmenes dinámicos

#### ■ Paridad:

- Cada vez que se escriben datos, se calcula el XOR bit a bit (1 número de unos impar, 0: número de unos par) de los bloques implicados en cada disco.
- Basada en operaciones XOR:

Disco 1:	00101010	(Datos)
Disco 2:	10001110	(Datos)
Disco 3:	11110111	(Datos)
Disco 4:	10110101	(Datos)
Disco 5:	11100110	(Paridad)

### Administración de volúmenes dinámicos

#### ■ Paridad:

- Si uno de los discos falla (p.ej. el disco 4), el contenido se puede restaurar a partir de la paridad:

Disco 1:	00101010	(Datos)
Disco 2:	10001110	(Datos)
Disco 3:	11110111	(Datos)
Disco 5:	11100110	(Paridad)
Disco 4:	10110101	(Datos)

### Administración de volúmenes dinámicos

#### ■ Control de dispositivos de entrada/salida:

- La herramienta `mdadm` permite crear o administrar un dispositivo RAID, convertir un disco “normal” en un RAID...
- Tiene distintos modos de funcionamiento:
  - `create`: configurar y activar sistemas RAID.
- `/proc/mdstat` lista todos los sistemas RAID (dispositivos `md`) activos con información sobre su estado.
- Las particiones que formen el RAID tienen que un *flag* RAID (*Linux raid auto*), de esta manera serán detectadas y activadas en el proceso de arranque.

### Administración de volúmenes dinámicos

#### ■ Ejemplo de creación de un RAID1.

- Instalar el paquete `mdadm`
- Activar el *flag* RAID en la partición (tipo `fd` en `fdisk`, *raid* en `gparted`).
- Crear el RAID (necesario instalar paquete `mdadm`):

```
1 mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sda1 /dev/sdc1
```

- Crear un SF sobre el sistema RAID: `mke2fs -t ext4 /dev/md1`.
- Añadirlo al fichero `/etc/fstab` para montarlo en tiempo de arranque:

```
1 /dev/md1 /home ext4 defaults 1 2
```

### Administración de volúmenes dinámicos

#### ■ Creación de un RAID1 con un disco que ya tiene datos:

- Crear el RAID con la partición que tiene los datos:

```
1 mdadm --create /dev/md2 --force --level=1 --raid-devices=1 /dev/sda4
```

- Añadir nuevo disco al RAID como disco de repuesto (*spare*):

```
1 mdadm /dev/md2 -a /dev/sdc3
```

- Activar el nuevo disco: `mdadm --grow /dev/md2 -n 2`.
- A continuación, introducirlo en `/etc/fstab`.

#### ■ Información sobre el estado:

```
1 mdadm --detail --scan /dev/md1
```

- Todo esto se puede configurar utilizando el fichero `/etc/mdadm.conf`.

## 6. Referencias

### Referencias

### Referencias

[Nemeth et al., 2010] Evi Nemeth, Garth Snyder, Trent R. Hein y Ben Whaley. Unix and Linux system administration handbook.

Capítulo 8. *Storage and disk*. Prentice Hall. Cuarta edición. 2010.

[Frisch, 2002] Aeleen Frisch. Essential system administration.

Capítulo 10. *Filesystems and disks*. O'Reilly and Associates. Tercera edición. 2002.