

Mantenimiento de los sistemas de archivos Linux

Los principales sistemas de archivos que constituyen la arborescencia global del sistema de archivos Linux casi siempre son creados durante la instalación del sistema. Sin embargo, el administrador a veces tiene que definir nuevos, cuando por ejemplo se añade hardware o nuevos recursos compartidos en red. Debe seleccionar el tipo de sistema de archivos, según las distintas limitaciones de explotación y los objetivos de utilización (rendimiento, flexibilidad de evolución, fiabilidad, tamaño máximo de los datos que se tienen que gestionar, tipo de material, cantidad de memoria viva, etc.).

Por otro lado, el administrador debe vigilar el estado de los recursos de almacenaje y optimizar su configuración en función de las estadísticas generales de uso.

Para ello dispone de numerosos comandos, unos genéricos y otros específicos para el tipo de sistema de archivos.

1. Creación y control de los sistemas de archivos

El comando `mkfs` permite crear un sistema de archivos especificando el tipo. Se trata de un comando genérico que hace una llamada a diferentes comandos en función del tipo de sistema de archivos especificado.

El comando `fsck` también es un comando genérico que permite verificar un sistema de archivos físico de cualquier tipo, siempre y cuando esté desmontado. Se ejecuta automáticamente cuando arranca el sistema, para todos los sistemas de archivos declarados en `/etc/fstab` (o para una unidad de montaje de sistema) con la opción de verificación activada.

a. Creación de un sistema de archivos: `mkfs`

La creación del sistema de archivos se hace con el comando `mkfs`.

Sintaxis

`mkfs [-t tipo] [OpcionesFS] ArchivoEspecial [NumeroBloques]`

Parámetros principales

<code>-t tipo</code>	Tipo de sistema de archivos que se creará.
<code>OpcionesFS</code>	Opciones de creación, específicas del tipo de sistema de archivos.
<code>ArchivoEspecial</code>	Espacio de almacenamiento en el que se creará el sistema de archivos.
<code>NumeroBloques</code>	Tamaño en bloques del sistema de archivos.

Descripción

Este comando genérico permite crear un sistema de archivos de cualquier tipo gestionado por el sistema. Sin la opción `-t`, el comando usa el tipo de sistema de archivos por defecto del sistema.

Según el tipo de sistema de archivos deseado, el comando hará una llamada al comando específico de creación de sistema de archivos de ese tipo, pasándole eventualmente las opciones especificadas mediante el argumento `OpcionesFS`. Si el tamaño no se ha especificado, el sistema de archivos será creado con una tamaño correspondiente al del espacio de almacenamiento.



Si un sistema de archivos ya existe, el comando solicita la confirmación para sobrescribirlo.

Ejemplo

Creación de un sistema de archivos ext4, con los atributos por defecto, en un espacio de almacenamiento. Como este último ya contiene un sistema de archivos, el comando solicita la confirmación para sobrescribirlo:

```
mkfs -t ext4 /dev/sdb1
mke2fs 1.44.5 (15-Dec-2018)
/dev/sdb1 contains a vfat file system
Proceed anyway? (y,N) y
Creating filesystem with 203776 1k blocks and 51000 inodes
Filesystem UUID: 47c46533-94ce-4738-a229-8455caf5a930
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Se comprueba:

```
blkid /dev/sdb1
/dev/sdb1: UUID="47c46533-94ce-4738-a229-8455caf5a930" TYPE="ext4"
PARTUUID="804d16f1-01"
```

b. Comprobación de los sistemas de archivos

Comprobar un sistema de archivos consiste en controlar la coherencia de las estructuras de gestión de su contenido, los metadatos y, si es posible, corregirlo en caso de problema. En particular, hay que asegurarse de que la tabla de los inodos (archivos y directorios creados en el sistema de archivos) sea coherente con la tabla de bloques de datos atribuidos a los archivos y directorios. Si no es el caso, la herramienta de verificación podrá asociar los bloques "perdidos" a los archivos creados especialmente a ese efecto.

Para los sistemas de archivos transaccionales (journaling), la comprobación por defecto se apoya en el registro de transacciones del sistema de archivos, lo que permite determinar las eventuales operaciones interrumpidas antes de ejecutarlas correctamente. Los riesgos de pérdida de coherencia son, por lo tanto, limitados y la comprobación es

muy rápida, ya que ya no es necesario comprobar todos los bloques atribuidos.

La comprobación del sistema de archivos se puede hacer con el comando genérico `fsck`. El sistema de archivos no tiene que estar montado.

Sintaxis

```
fsck -t tipo ArchivoEspecial
```

Parámetros principales

<code>-t tipo</code>	Tipo de sistema de archivos para verificar.
<code>-f</code>	Verificación completa del sistema de archivos transaccional.
<code>ArchivoEspecial</code>	Sistema de archivos para verificar.

Descripción

Para los sistemas de archivos transaccionales, la opción `-f` fuerza una verificación completa de un sistema de archivos, sin limitarse a los elementos del registro de transacciones.



Incluso si el comando `fsck` gestiona los sistemas de archivos de tipo XFS, es preferible usar los comandos específicos `xfs_check` y `xfs_repair`.

Ejemplo

Comprobación del sistema de archivos creado en el ejemplo anterior:

fsck /dev/sdb1

fsck de util-linux 2.33.1

e2fsck 1.44.5 (15-Dec-2018)

/dev/sdb1: clean, 11/51000 files, 12090/203776 blocks

Como el sistema de archivos es de tipo ext4, transaccional, la comprobación ha sido rápida, ya que solo utiliza los datos transaccionales. Se puede forzar una verificación completa con la opción **-f** :

fsck -f /dev/sdb1

fsck de util-linux 2.33.1

e2fsck 1.44.5 (15-Dec-2018)

Pass 1: Checking inodes, blocks, and sizes

Pass 2: Checking directory structure

Pass 3: Checking directory connectivity

Pass 4: Checking reference counts

Pass 5: Checking group summary information

/dev/sdb1: 11/51000 files (0.0% non-contiguous), 12090/203776 blocks

Se puede intentar comprobar el sistema de archivos **/dev/sdb1** , montado:

fsck /dev/sdb1

fsck de util-linux 2.33.1

e2fsck 1.44.5 (15-Dec-2018)

/dev/sdb1 is mounted.

e2fsck: Cannot continue, aborting.

2. Creación de una zona de swap

Una zona de swap se define en el momento de la instalación del sistema, con la creación del espacio de swap correspondiente. Sin embargo, el administrador puede modificar posteriormente la configuración de la zona de swap, cambiando el espacio de swap o añadiendo nuevos espacios de swap, de manera puntual o permanente.

a. Elección de los espacios de swap

Se ha visto que en los sistemas modernos, el mecanismo de swap se utiliza poco: la memoria física y el espacio de disco ya no son recursos caros y las configuraciones de hardware de los servidores están generalmente sobredimensionadas.

A pesar de ello, es posible que haya picos de uso de la memoria viva, que pueden provocar que el sistema consuma swap (por ejemplo, en un pequeño servidor de desarrollo, una compilación compleja o el lanzamiento de una aplicación que necesite mucha memoria, en un servidor de red, un crecimiento brusco del número de conexiones simultáneas o de solicitudes complejas). Para evitar el bloqueo del sistema o incluso su paro en modo *"panic"*, la configuración de la zona de swap debe ser óptima. El administrador debe también poder añadir puntualmente espacios de swap en el caso de una saturación de la memoria viva.

La eficacia del swap depende del tamaño de la zona de swap con respecto al de la memoria viva usada por las aplicaciones, y de los tiempos de acceso a esos espacios de almacenamiento.

b. Tipos de espacio de swap

La zona de swap puede estar compuesta por uno o distintos espacios de swap, de tipos diferentes:

- ▾ Partición o volumen lógico dedicado

Es el más frecuente. La partición o el volumen lógico debe ser de un tipo particular (82, swap). Su contenido no está estructurado en directorios y archivos, no existe la operación de montaje ya que las aplicaciones no pueden acceder directamente. El gestor de swap administra su contenido.

- ▾ Archivo

Se puede añadir a la zona de swap un archivo de tamaño fijo. Esta solución no es óptima ya que los tiempos de acceso se encuentran degradados, ya que el gestor de swap utiliza los pilotos de los sistemas de archivos para acceder al propio archivo. Normalmente, esta solución se aplica como último recurso y de manera temporal.

La configuración de una zona de swap multiespacio permite especificar las prioridades de

uso entre esos mismos espacios. De esta manera se puede privilegiar a los que tengan un buen tiempo de acceso, por ejemplo en este orden: memoria viva, discos SSD, disco locales rápidos dedicados, SAN y archivos.

c. Creación de un espacio de swap: mkswap

Un espacio de swap debe estar "formateado" específicamente con el comando `mkswap`, para poder ser utilizado.

Sintaxis

```
mkswap espacio_almacenaje
```

Parámetros principales

<code>/camino/archivo</code>	Estructura el archivo para que pueda ser explotado como espacio de swap.
<code>/dev/Archivo_Especial</code>	Estructura el espacio de almacenamiento designado por el archivo especial para que pueda ser explotado como espacio de swap.
<code>-L LABEL</code>	Estructura el espacio de almacenamiento designado por la etiqueta <code>LABEL</code> para que pueda ser explotado como espacio de swap.
<code>-U UUID</code>	Estructura el espacio de almacenamiento designado por el UUID <code>UUID</code> para que pueda ser explotado como espacio de swap.

Ejemplo

En caso de urgencia, se puede utilizar un archivo como espacio de swap. Primero hay que

crear un archivo del tamaño deseado:

```
fallocate --length 1GiB /var/miswap  
ls -l /var/miswap  
-rw-r--r-- 1 root root 1073741824 jul 30 23:58 /var/miswap
```

Es prudente cambiar los derechos de acceso al archivo:

```
chmod 0600 /var/miswap
```

Después hay que formatear el contenido en espacio de swap:

```
mkswap /var/miswap  
Configurando espacio de intercambio versión 1, tamaño = 1024 MiB (1073737728 bytes)  
sin etiqueta, UUID=e60e394e-cf08-4e27-88bd-d5b914866bbc
```

Después se puede activar ese espacio para integrarlo en la zona de swap del sistema:

```
swapon /var/miswap
```

Se comprueba:

```
swapon -s
```

Nombre del fichero	Tipo	Tamaño	Utilizado	Prioridad
/dev/sda5	partition	998396	0	-2
/var/miswap	file	1048572	0	-3

Esta operación solo debería ser temporal, porque el tiempo de acceso a un archivo de swap es mucho más lento que el tiempo de acceso una partición de swap. En el próximo arranque, el archivo de swap no será activado, excepto si se declara en el archivo `/etc/fstab`.

d. Control de la zona de swap

El comando `swapon` con la opción `-s` permite visualizar la configuración de la zona de swap.

También se pueden utilizar las herramientas de seguimiento de los recursos para supervisar el consumo de la memoria viva, el comando `top` por ejemplo.

3. Los principales tipos de sistemas de archivos en Linux

Linux soporta numerosos tipos de sistemas de archivos. La elección del tipo de sistema de archivos depende de la naturaleza del soporte físico, de sus capacidades, así como del tipo de uso (lectura, escritura, archivos voluminosos pero poco numerosos, muchos archivos pequeños ...) y de las funcionalidades deseadas (velocidad de acceso en lectura y/o escritura, seguridad, atributos gestionados, número y tamaño de los objetos almacenados).

A continuación se van a describir los principales tipos de sistema de archivos utilizados en Linux, los cuales tenemos que conocer para la certificación.

a. Sistemas de archivos de tipo ext*

El tipo ext ha evolucionado en distintas versiones a lo largo del tiempo: ext2, ext3 y ext4.

ext2

Los sistemas de archivos de tipo ext2 han sido creados especialmente para Linux. Durante mucho tiempo han sido el tipo de sistema de archivos por defecto en la mayoría de las distribuciones. Permiten gestionar archivos de un tamaño máximo de 2 TB, en volúmenes de 32 TB como máximo. Las fechas de los archivos están soportadas hasta 2038.

Este tipo de sistema de archivos evolucionó hacia ext3 y después ext4, ya casi no se usa hoy en día, excepto por la compatibilidad con las infraestructuras existentes. Su principal inconveniente es que no implementa el registro por diario, lo que puede provocar una

alteración de la estructura del mismo sistema de archivos en caso de interrupción brutal durante una modificación, convirtiéndolo en parcial o totalmente inutilizable.

ext3

Se trata de una evolución del tipo de sistema de archivos ext2, que permite implementar un registro por diario de transacciones para asegurar la coherencia de los metadatos del sistema de archivos en caso de interrupción brutal de las escrituras.

Cada operación de escritura está inscrita previamente en el diario, y validada después en ese mismo diario. En caso de problema durante la operación, se puede anular o volver a ejecutar el conjunto de la transacción de escritura: incluso en el caso de que se perdieran los datos que se iban a escribir, el sistema de archivos guardaría una coherencia.

Este mecanismo de registro de diario permite también acelerar las operaciones de comprobación del sistema de archivos, ya que basta con consultar el diario para encontrar las transacciones potencialmente interrumpidas.

El registro del diario es opcional. Un sistema de archivos de tipo ext3 es compatible con ext2 en lectura y escritura.

Los límites de tamaño de los archivos y de los volúmenes son los del tipo ext2, así como las fechas para los archivos.

ext4

Este nuevo tipo de sistema de archivos está destinado a remplazar a ext3. Permite gestionar archivos de 16 TB y volúmenes de 1 EiB. Una nueva técnica de gestión del espacio asignado a los archivos, los *extents*, permite limitar considerablemente la fragmentación. Las fechas de los archivos están soportadas hasta 2514.

Un sistema de archivos de tipo ext3 es compatible con ext4 en lectura y escritura.

Es el tipo de sistema de archivos que se encuentra por defecto en numerosas distribuciones, en particular Debian.

b. Sistemas de archivos de tipo XFS

XFS es un sistema de archivos concebido por la empresa SGI (*Silicon Graphics Inc*) para los sistemas Unix IRIX. Creado en 1994, disfruta de una experiencia muy importante. Es

utilizado, principalmente, en entornos de producción, con servidores que administran una cantidad muy grande de almacenaje de disco.

SGI le ha dado a este sistema de gestión de archivos una licencia GPL en el año 2000, lo que permitió la portabilidad a Linux. Ha sido integrado en el núcleo a partir de la versión 2.4, está disponible en la mayoría de las distribuciones.

Desarrollado, especialmente, para gestionar de manera eficaz grandes volúmenes de disco, sus principales características son las siguientes:

- ✧ 64 bits: el límite máximo teórico para el tamaño de un archivo o de un sistema de archivos completo es de 8 exabytes ($2^{63} - 1$). En la práctica, este tamaño está limitado por parámetros relativos a cada distribución (100 TB para Red Hat 6, 500 TB para Red Hat 7), pero permite superar con mucho margen los 16 TB, tamaño límite del sistema de archivos de tipo ext.
- ✧ Registro: las operaciones de escritura en los metadatos se describen en el registro antes de ser ejecutadas, lo que permite garantizar la coherencia del sistema de archivos en caso de avería de tipo material. Para optimizar el rendimiento, es posible posicionar el diario en un volumen físico diferente con respecto a donde se encuentran los datos.
- ✧ E/S paralelas: gracias a su organización en **grupos de asignación** (internamente, un sistema de archivos se divide en diferentes subconjuntos independientes), permite efectuar simultáneamente distintas operaciones de entrada/salida en un mismo sistema de archivos, lo que está particularmente adaptado para las aplicaciones de tipo multiproceso.
- ✧ Diferentes técnicas permiten asegurar un alto nivel de rendimiento y de evolución: uso de árboles indexados (**B-tree**) para los metadatos, asignación diferida, preasignación para evitar la fragmentación, desfragmentación en línea, asignación por extensiones, etc.
- ✧ Las herramientas de comprobación y de corrección de los sistemas de archivos son muy rápidas.
- ✧ Atributos extendidos: permiten gestionar **listas de control de acceso** (ACL), además de los derechos de acceso clásicos del mundo Unix/Linux, así como atributos específicos para algunas aplicaciones.
- ✧ Sistema de cuotas de uso del espacio muy flexible, por grupo, usuario o "proyecto" (cuotas asociadas a una rama de la arborescencia).

- ✓ Gestión de **instantáneas** (*snapshots*): esta técnica permite la toma de una imagen del sistema de archivos en un instante preciso y, por lo tanto, permite efectuar respaldos en línea.
- ✓ Extensible en línea: un sistema de archivos puede ser extendido durante su uso (pero no reducido).
- ✓ Herramientas de respaldo/restauración específicas para el sistema de archivos.

Este sistema de gestión de archivos está, sobre todo, adaptado para grandes servidores, necesita mucha memoria viva e implica el uso de material fiable y seguro (caché asegurada por una batería).

XFS es el sistema de archivos presente por defecto en las distribuciones de tipo Red Hat, a partir de la versión 7.

c. Sistemas de archivos de tipo Btrfs

Btrfs (*B-tree File System*, se pronuncia "báterefes") es el más reciente de los sistemas de archivos creados para Linux. Implementa conceptos modernos para ofrecer un conjunto de funcionalidades de alto nivel, fruto de la colaboración de grandes actores del mundo Linux (Red Hat, Oracle, SUSE, Intel, etc.).

Se trata de un sistema de gestión de archivos de 64 bits, usando una técnica de **copy on write (CoW)**: para garantizar la integridad de los datos, una modificación se hace copiando los datos de origen y después modificando la copia), implementando capacidades de compresión, de respaldo en línea, de redundancia de datos, de instantáneas, de redimensionamiento en línea, etc.



Integrado en el núcleo desde 2008, estaba previsto que se convirtiera a largo plazo en el sistema de archivos por defecto de Linux, pero no está soportado en todas las distribuciones. Red Hat decidió dejar de soportarlo a partir de la versión 8 de su distribución RHEL, al contrario que SUSE que lo utiliza por defecto para el sistema de archivos raíz. Algunas funcionalidades avanzadas todavía no son estables.

Sus principales puntos fuertes son los siguientes:

- ˘ Sistema de archivos de 64 bits: tamaño máximo de un sistema de archivos 16 exabytes y tamaño máximo de un archivo 8 exabytes.
- ˘ Optimización de la gestión de los archivos pequeños.
- ˘ Consideración de las particularidades de los discos SSD.
- ˘ Fiabilidad aumentada por checksum de los datos y metadatos.
- ˘ Gestión de las instantáneas (snapshots) en lectura y lectura/escritura.
- ˘ Gestión de subvolúmenes.
- ˘ Algoritmos optimizados de compresión.
- ˘ Desfragmentación en línea.
- ˘ Gestión integrada del multivolumen (RAID).



Para la certificación necesita conocer las características principales de este tipo de sistema de archivos y ser capaz de gestionar una implementación simple, incluyendo subvolúmenes e instantáneas.

d. Sistemas de archivos de tipo ZFS

ZFS es un tipo de sistema de archivos, de **Sun Microsystems**, desarrollado para el sistema operativo Solaris, de tipo Unix. Sun lo implementó a partir de 2006, con una licencia de tipo software libre, lo que permitió su portabilidad hacia otros sistemas operativos. Oracle adquirió Sun Microsystems en 2010 y optó por una licencia propietaria para ZFS. El proyecto **OpenZFS** fue creado para mantener y hacer progresar una versión libre de ZFS.

Como consecuencia del conflicto de licencia, ZFS no ha sido integrado en el núcleo Linux. Para implementarlo en una plataforma Linux, hay que instalar un módulo del núcleo dinámico facilitado por el proyecto **Zol** (*ZFS on Linux*), desarrollado por el **Lawrence Livermore National Laboratory**.



A partir de la versión 8 de la distribución RHEL, ZFS ya no está soportado por Red Hat.

Este tipo de sistema de archivos implementa numerosas funcionalidades, integrando la gestión de volúmenes lógicos y de discos RAID. Permite gestionar de manera eficaz los volúmenes muy grandes de manera fiable y eficaz. Propone funcionalidades avanzadas, como las instantáneas (*snapshots*), la copia en escritura (*copy on write*) así como el control y la reparación automática en continuo.

ZFS gestiona el conjunto de los aspectos de almacenamiento, desde los dispositivos utilizados y combinados hasta la estructura interna del sistema de archivos. Eso le permite optimizar el almacenaje, la fiabilidad y el rendimiento de entradas/salidas tomando en cuenta los diferentes niveles (desde el punto de vista de hardware y software) implementados.

Propone funcionalidades de alto nivel, particularmente gracias a la gestión avanzada de las instantáneas (*snapshots*), que permiten el clonado o la marcha atrás hacia una versión anterior del estado del sistema de archivos (*check point rollback*).

Concebido para gestionar grandes volúmenes de datos, necesita sistemas de discos eficientes y servidores que dispongan de mucha memoria viva.

Sus principales puntos fuertes son los siguientes:

- ˘ Sistema de archivos de 128 bits, llevando los límites de capacidad más allá de las capacidades físicas de los sistemas actuales: tamaño máximo de un sistema de archivos 16 exabytes, tamaño máximo de un archivo 16 exabytes, número máximo de archivos por directorio 2^{48} , etc.
- ˘ Mecanismo de checksum jerárquico para garantizar los datos y los metadatos del sistema de archivos.
- ˘ Múltiples versiones de un conjunto de datos y metadatos (*snapshots*), que permiten gestionar restauraciones hacia un estado anterior.
- ˘ Reparación automática de las incoherencias o errores de escritura, a través del registro, la gestión de la paridad y la redundancia de datos (RAID).

- ˘ Gestión de software integrada de los diferentes niveles de RAID.
- ˘ Gestión de los dispositivos de caché.
- ˘ Gestión de las instantáneas (*snapshots*) para el respaldo y la replicación.
- ˘ Gestión de la compresión de datos.
- ˘ Gestión de la deduplicación de datos. Esta necesita de una gran cantidad de memoria viva.
- ˘ Reconstrucción rápida del sistema de archivos en caso de corrupción.
- ˘ Configuración sutil de los parámetros para la optimización.

4. Gestión de los sistemas de archivos de tipo ext2, ext3 y ext4

Un conjunto de herramientas específicas permite crear y modificar esos tipos de sistemas de archivos. Esos comandos se aplican en general a las tres versiones, ext 2, 3 o 4, incluso si la mayoría han conservado «e2» en su nombre. Pueden ser llamadas por los comandos genéricos que ya se han visto anteriormente o pueden ser utilizadas directamente.

a. Creación de un sistema de archivos ext*

El comando `mke2fs` permite crear sistemas de archivos ext2, 3 o 4.

Sintaxis

```
mke2fs [-j] [-t tipoFS] [-F] [-L label] [-U uuid] ArchivoEspecial
```

Parámetros principales

<code>-t tipoFS</code>	Tipo de sistema de archivos que se va a crear.
<code>-j</code>	Tipo de sistema de archivos que se va a crear: ext3.
<code>-F</code>	Fuerza la sobrescritura del sistema de archivos existente.
<code>-L label</code>	Etiqueta que se asignará al sistema de archivos.
<code>-U uuid</code>	UUID que se asignará al sistema de archivos.
<code>ArchivoEspecial</code>	Archivo especial asociado al espacio de almacenaje del sistema de archivos que se va a crear.

Descripción

La opción `-j` (*journal*) fuerza la creación de un sistema de archivos de tipo ext3. También se puede especificar el tipo del sistema de archivos (ext2, ext3 o ext4) con la opción `-t tipoFS`.

Sin una de esas dos opciones, el tipo de sistema de archivos usado por defecto será el configurado en el archivo `/etc/mke2fs.conf`.

Hay muchas opciones que permiten fijar las características detalladas del sistema de archivos (tamaño y número de bloques, número de inodos, ubicación del archivo de registro...).

El argumento `ArchivoEspecial` determina el espacio de almacenamiento usado para el sistema de archivos: archivo especial de tipo bloque, camino de red y volumen lógico LVM.

Ejemplos

Creación de un sistema de archivos ext3 en un volumen lógico :


```

mke2fs -j /dev/debian10-vg/lvol0
mke2fs 1.44.5 (15-Dec-2018)
/dev/debian10-vg/lvol0 contains a ext2 file system
    created on Sun Mar  1 21:56:32 2020
Proceed anyway? (y,N) y
Creating filesystem with 155648 1k blocks and 38912 inodes
Filesystem UUID: efd18095-e45f-44b4-b4b9-4f3ea7091b7a
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

```

El volumen lógico contenía ya un sistema de archivos, el comando ha solicitado la confirmación antes de sobrescribir.

Creación de un sistema de archivos ext4 en una partición de disco, con la opción `-F` para forzar la sobrescritura del sistema de archivos ya existente:

```

mke2fs -Ft ext4 /dev/sdb1
mke2fs 1.44.5 (15-Dec-2018)
/dev/sdb1 contains a ext2 file system
    created on Mon Mar  2 20:48:56 2020
Creating filesystem with 1976082 4k blocks and 494832 inodes
Filesystem UUID: 5513948e-e3e3-4764-8782-3875a0b09e1b
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

```

Comprobación:

```
blkid /dev/sdb1 /dev/debian10-vg/lvol0
/dev/sdb1: UUID="5513948e-e3e3-4764-8782-3875a0b09e1b" TYPE="ext4"
PARTUUID="c3072e18-01"
/dev/debian10-vg/lvol0: UUID="efd18095-e45f-44b4-b4b9-4f3ea7091b7a"
SEC_TYPE="ext2" TYPE="ext3"
```

b. Comprobación de un sistema de archivos ext*

El comando `e2fsck` permite comprobar y corregir, si fuera necesario, la estructura de un sistema de archivos ext*.

Sintaxis

```
e2fsck [-f] [-v] [-n | -p] ArchivoEspecial
```

Parámetros principales

<code>-f</code>	Fuerza el control completo del sistema de archivos (ext3, ext4).
<code>-v</code>	Visualización detallada.
<code>-n</code>	Control sin corrección.
<code>-p</code>	Corrección automática.
<code>ArchivoEspecial</code>	Archivo especial asociado al espacio de almacenamiento del sistema de archivos que se va a comprobar.

Descripción

El comando funciona en modo interactivo por defecto. Controla el sistema de archivos y, para cada anomalía constatada, propone una corrección que tiene que ser validada por el

usuario.

Si el sistema de archivos tiene un registro (ext3 o ext4), el comando solamente controla los elementos correspondientes en las transacciones indicadas en curso en el registro. La opción `-f` permite forzar un control completo del sistema de archivos, más allá del registro.

La opción `-n` provoca un control no interactivo, sin corrección.

Por otra parte, la opción `-p` provoca un control con corrección automático y no interactivo.



Se recomienda encarecidamente ejecutar este comando en un sistema de archivos desmontado.

c. Gestión de los parámetros de un sistema de archivos con tune2fs

El comando `tune2fs`, que ya se vio anteriormente para gestionar la etiqueta y el UUID, permite mostrar el valor de numerosos parámetros de un sistema de archivos ext* y modificar algunos de ellos.

Sintaxis

```
tune2fs [-l] [-j] [-J OpcionesRegistro] [-E OpcionesExtendidas]
[-L label] [-U uuid] ArchivoEspecial
```

Paramètres principaux

<code>-l</code>	Lista las características del sistema de archivos.
<code>-j</code>	Crea un registro para el sistema de archivos.
<code>-J</code> <code>OpcionesRegistro</code>	Modifica las opciones del registro.
<code>-E</code> <code>OpcionesExtendidas</code>	Modifica las opciones extendidas del sistema de archivos.
<code>-L label</code>	Etiqueta que se asignará al sistema de archivos.
<code>-U uuid</code>	UUID que se asignará al sistema de archivos.
<code>ArchivoEspecial</code>	Archivo especial asociado al espacio de almacenamiento del sistema de archivos.

Descripción

Este comando permite mostrar y modificar numerosos parámetros de un sistema de archivos ext *. En particular, permite la creación de un registro (opción `-j`) para un sistema de archivos de tipo ext2, lo que lo transforma en tipo ext3.

Ejemplos

Muestra los parámetros de un sistema de archivos en ext4:

```
tune2fs -l /dev/sdb1
tune2fs 1.44.5 (15-Dec-2018)
Filesystem volume name: <none>
Last mounted on:      <not available>
Filesystem UUID:      5513948e-e3e3-4764-8782-3875a0b09e1b
Filesystem magic number: 0xEF53
```

```

Filesystem revision #: 1 (dynamic)
Filesystem features:  has_journal ext_attr resize_inode dir_index
filetype extent 64bit flex_bg sparse_super large_file huge_file dir_nlink
extra_isize metadata_csum
Filesystem flags:     signed_directory_hash
Default mount options: user_xattr acl
Filesystem state:     clean
Errors behavior:      Continue
Filesystem OS type:   Linux
Inode count:          494832
Block count:          1976082
Reserved block count: 98804
Free blocks:          1919949
Free inodes:           494821
First block:           0
Block size:            4096
Fragment size:         4096
Group descriptor size: 64
Reserved GDT blocks:   964
Blocks per group:      32768
Fragments per group:   32768
Inodes per group:      8112
Inode blocks per group: 507
Flex block group size: 16
Filesystem created:    Mon Mar 2 20:52:11 2020
Last mount time:       n/a
Last write time:       Mon Mar 2 20:52:12 2020
Mount count:           0
Maximum mount count:   -1
Last checked:          Mon Mar 2 20:52:11 2020
Check interval:        0 (<none>)
Lifetime writes:       68 MB
Reserved blocks uid:   0 (user root)
Reserved blocks gid:   0 (group root)
First inode:           11
Inode size:            256
Required extra isize:  32
Desired extra isize:   32
Journal inode:         8
Default directory hash: half_md4

```

```

Directory Hash Seed:  f687383b-2d33-40b1-abc6-9afc940dce23
Journal backup:      inode blocks
Checksum type:       crc32c
Checksum:            0xc65bb829

```

En la sección `Filesystem features`, podemos notar la existencia de un registro (`has_journal`), de gestión de los extents (`extent`) y de los archivos voluminosos (`huge_file`), características de un tipo ext4.

Conversión de un sistema de archivos de tipo ext2 a tipo ext3:

```

blkid /dev/debian10-vg/lvol1
/dev/debian10-vg/lvol1: UUID="39283d6c-e6f8-44cc-aa57-
46ef454523bf" TYPE="ext2"

```

Se convierte en ext3:

```

tune2fs -j /dev/debian10-vg/lvol1
tune2fs 1.44.5 (15-Dec-2018)
Creating journal inode: done

```

Se comprueba:

```

blkid /dev/debian10-vg/lvol1
/dev/debian10-vg/lvol1: UUID="39283d6c-e6f8-44cc-aa57-46ef454523bf"
SEC_TYPE="ext2" TYPE="ext3"

```

d. Depuración de un sistema de archivos ext*

El comando `debugfs` permite comprobar el estado de un sistema de archivos ext* en modo de depuración.

[Sintaxis](#)

debugfs [-w] ArchivoEspecial

Parámetros principales

<code>-w</code>	Modo escritura.
<code>ArchivoEspecial</code>	Archivo especial asociado al espacio de almacenamiento del sistema de archivos que se quiere depurar.

Descripción

Este comando funciona en modo interactivo. Por defecto, está en modo de solo lectura y no permite modificar el sistema de archivos. La opción `-w` autoriza la modificación.

La herramienta abre el sistema de archivos y muestra una línea de comandos. Numerosos comandos permiten manipular las estructuras y objetos del sistema de archivos, entre ellos:

▾ `help`

Ayuda.

▾ `quit`

Cierra `debugfs`.

▾ `logdump`

Muestra el contenido del registro.

▾ `show_super_stats`

Muestra el contenido del super bloque y de los descriptores de grupos de bloques.

▾ `cat archivo`

Muestra el contenido de `archivo`.

```
stat archivo
```

Muestra el contenido de la entrada de la tabla de los inodos para `archivo`.

```
icheck block ...
```

Lista los inodos utilizando los bloques específicos.

```
set_super_value campo valor
```

Modifica el contenido del campo especificado del super bloque.

```
undel numeroInode
```

Declara activo el inodo de número `numeroInode`.

```
free archivo
```

Libera el inodo `archivo`.

```
kill_file archivo
```

Suprime `archivo` y libera sus bloques de datos.

```
rmdir dir
```

Suprime el directorio `dir`.

```
unlink CaminoArchivo
```

Suprime el enlace entre `CaminoArchivo` y su inodo.

Ejemplo

Comprobación, en modo depuración, de un sistema de archivos ext4:

debugfs /dev/mapper/debian10--vg-home

debugfs 1.44.5 (15-Dec-2018)

debugfs:

Se lista el registro de transacciones:

debugfs: **logdump**

Journal starts at block 1, transaction 83

Found expected sequence 83, type 1 (descriptor block) at block 1

Found expected sequence 83, type 2 (commit block) at block 4

Found expected sequence 84, type 1 (descriptor block) at block 5

Found expected sequence 84, type 2 (commit block) at block 7

Found expected sequence 85, type 2 (commit block) at block 8

Found expected sequence 86, type 1 (descriptor block) at block 9

Found expected sequence 86, type 2 (commit block) at block 11

Found expected sequence 87, type 1 (descriptor block) at block 12

Found expected sequence 87, type 2 (commit block) at block 14

Found expected sequence 88, type 1 (descriptor block) at block 15

Found expected sequence 88, type 2 (commit block) at block 17

No magic number at block 18: end of journal.

debugfs:

Se han validado todas las transacciones (commit).

Se comprueban los atributos de la raíz:

debugfs: **stat /**

Inode: 2 Type: directory Mode: 0755 Flags: 0x80000

Generation: 0 Version: 0x00000000:00000001

User: 0 Group: 0 Project: 0 Size: 4096

File ACL: 0

Links: 4 Blockcount: 8

Fragment: Address: 0 Number: 0 Size: 0

ctime: 0x5e598c14:8d137a80 -- Fri Feb 28 22:54:28 2020

atime: 0x5e598c1a:4d2e0e80 -- Fri Feb 28 22:54:34 2020

mtime: 0x5e598c14:8d137a80 -- Fri Feb 28 22:54:28 2020

crttime: 0x5e594a21:00000000 -- Fri Feb 28 18:13:05 2020

Size of extra inode fields: 32
Inode **checksum**: 0x2af31a6b
EXTENTS:
(0):9240

El directorio raíz del sistema de archivos posee el número de inodo **2**. Sus permisos son **0755**, pertenece al usuario root (UID = 0). Tiene cuatro enlaces físicos.

Se sale de la herramienta:

debugfs: **quit**

e. Respaldo del sistema de archivos ext*

El comando **dump** gestiona el respaldo de un sistema de archivos de tipo ext*. El respaldo puede ser total o incremental.

Sintaxis

dump [-nivel#] [-f ArchivoRespaldo] [-v] [-u] SistArchivo

Parámetros principales

<code>-nivel#</code>	Nivel de respaldo (de 0 a n).
<code>-f</code> <code>ArchivoRespaldo</code>	Archivo o periférico de respaldo.
<code>-v</code>	Visualización detallada.
<code>-u</code>	Actualización del archivo de fechas de respaldo <code>/etc/dumpdates</code> .
<code>SistArchivo</code>	Sistema de archivos que se va a respaldar.

Descripción

El comando gestiona diferentes niveles de respaldo, numerados de `0` a `n`, `0` indica un respaldo total. La información (nivel, fecha) de los respaldos efectuados se almacena por defecto en el archivo `/etc/dumpdates`.

El sistema de archivos que se quiera respaldar puede ser especificado con un archivo especial o se puede especificar con su punto de montaje.

El sistema de archivos puede estar montado o no.

Ejemplo

Ejemplo de respaldo total de un sistema de archivos ext4 con `dump`:

```
dump -0 -f /root/svg /dev/mapper/vg--datos-lvol1
DUMP: Date of this level 0 dump: Wed Mar  4 19:45:06 2020
DUMP: Dumping /dev/mapper/vg--datos-lvol1 (an unlisted file system) to /root/svg
DUMP: Label: none
DUMP: Writing 10 Kilobyte records
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
```

```

DUMP: estimated 12106 blocks.
DUMP: Volume 1 started with block 1 at: Wed Mar  4 19:45:06 2020
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing /root/svg
DUMP: Volume 1 completed at: Wed Mar  4 19:45:06 2020
DUMP: Volume 1 12830 blocks (12.53MB)
DUMP: 12830 blocks (12.53MB) on 1 volume(s)
DUMP: finished in less than a second
DUMP: Date of this level 0 dump: Wed Mar  4 19:45:06 2020
DUMP: Date this dump completed: Wed Mar  4 19:45:06 2020
DUMP: Average transfer rate: 0 kB/s
DUMP: DUMP IS DONE
ls -l /root/svg
-rw-r--r--. 1 root root 13137920 4 marzo 19:46 /root/svg

```

En este ejemplo, `0` es el nivel del respaldo (0 para un respaldo completo, *n* para cada número que se incrementa), la opción `-f` indica el archivo de destino del respaldo y, finalmente, el último parámetro es el volumen lógico del sistema de archivos que se va a respaldar.

f. Restauración de un sistema de archivos ext*

El comando `restore` gestiona la restauración de un sistema de archivos ext*.

Sintaxis

```

restore -r -f ArchivoRespaldo
restore -x -f ArchivoRespaldo [dir]
restore -t -f ArchivoRespaldo [archivos]
restore -C -f ArchivoRespaldo
restore -i -f ArchivoRespaldo

```

Parámetros principales

<code>-f</code>	Archivo o periférico de respaldo.
<code>ArchivoRespaldo</code>	
<code>-r</code>	Restauración del sistema de archivos.
<code>-x</code>	Restauración de los directorios <code>dir</code> del sistema de <code>archivos</code> .
<code>-t</code>	Lista el contenido del respaldo o los elementos de archivos.
<code>-C</code>	Compara el respaldo con el sistema de archivos actual.
<code>-i</code>	Restauración en modo interactivo.

Descripción

Este comando permite gestionar un respaldo del sistema de archivos ext* efectuado con el comando `dump`. Se puede listar todo su contenido o una parte (opción `-t`), comparar los archivos y los directorios respaldados con respecto a los del sistema montado (opción `-C`), restaurar todo el sistema de archivos o una parte (opciones `-r` o `-x`), o trabajar en modo interactivo (opción `-i`).

La restauración se hace a partir del directorio actual, por lo tanto hay que desplazarse a la raíz del sistema de archivos destinatario.

Ejemplo

Restauración de un sistema de archivos ext4:

```
cd /var/datos/rest
restore -rvf /root/svg
Verify tape and initialize maps
Input is from a local file/pipe
```

```

Input block size is 32
Dump  date: Wed Mar  4 19:46:23 2020
Dumped from: the epoch
Level 0 dump of /mnt on pbaCOS8:/dev/mapper/vg--datos-lvol1
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Make node ./lost+found
Make node ./pba
Make node ./pba/.config
[...]
extract file ./pba/.local/share/recently-used.xbel
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore

```

5. Gestión de base de los sistemas de archivos de tipo Btrfs

Para configurar y gestionar un sistema de archivos de tipo Btrfs, hay que usar comandos específicos, proporcionados por el paquete `btrfs-tools`.

Se verán los diferentes comandos que permiten crear un sistema de archivos Btrfs, comprobar su coherencia y sus parámetros, y cómo crear instantáneas (*snapshots*) y subvolúmenes.

a. Creación de un sistema de archivos Btrfs

El comando `mkfs.btrfs` permite crear un sistema de archivos Btrfs.

Sintaxis

`mkfs.btrfs [Opciones] ArchivoEspecial`

Parámetros principales

Opciones	Opciones de creación.
ArchivoEspecial	Camino de acceso del espacio de almacenaje.

Descripción

Se pueden especificar numerosas opciones, en particular:

`-f o --force`

Fuerza la sobrescritura del sistema de archivos existente.

`-L o --label label`

Atribuye la etiqueta `label` al sistema de archivos.

`-U o --uuid uuid`

Atribuye el UUID `uuid` al sistema de archivos.

Ejemplo

Creación de un sistema de archivos Btrfs en una partición, con posible sobrescritura y usando los valores por defecto, en una distribución Debian 10:

Se instala el paquete:

apt-get install btrfs-tools

Leyendo lista de paquetes... Hecho

Creando árbol de dependencias

Leyendo la información de estado... Hecho

Nota, seleccionando «btrfs-progs» en lugar de «btrfs-tools»

Se instalarán los siguientes paquetes adicionales:

liblzo2-2

Paquetes sugeridos:

duperemove

Se instalarán los siguientes paquetes NUEVOS:

btrfs-progs liblzo2-2

0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.

Se necesita descargar 658 kB de archivos.

Se utilizarán 3.558 kB de espacio de disco adicional después de esta operación.

¿Desea continuar? [S/n] s

[...]

mkfs.btrfs -f /dev/sdb1

btrfs-progs v4.20.1

See <http://btrfs.wiki.kernel.org> for more information.

Label: (null)

UUID: 0ea1173f-9ee5-421e-ac7d-0cb374df8738

Node size: 16384

Sector size: 4096

Filesystem size: 7.54GiB

Block group profiles:

Data: single 8.00MiB

Metadata: DUP 385.94MiB

System: DUP 8.00MiB

SSD detected: no

Incompat features: extref, skinny-metadata

Number of devices: 1

Devices:

ID SIZE PATH

1 7.54GiB /dev/sdb1

El sistema de archivos ha sido creado.

b. Conversión de sistema de archivos ext* a Btrfs

Otra posibilidad sería convertir un sistema de archivos ext* en sistema de archivos Btrfs, con el comando `btrfs-convert`.

Sintaxis

```
btrfs-convert [Opciones] ArchivoEspecial
```

Parámetros principales

Opciones	Opciones de conversión.
Archivo Especial	Camino de acceso del espacio de almacenamiento.

Descripción

Esta operación es peligrosa y podría corromper el sistema de archivos existente. Por lo tanto, se aconseja encarecidamente hacer un respaldo antes de ejecutar la operación y comprobar la coherencia del sistema de archivos. El sistema de archivos debería ser desmontado antes de la conversión.

En caso de problema, se puede usar la opción `-r` (*rollback*) para anular la conversión.

La conversión se hace en el sistema de archivos existente, por lo tanto, no debe estar saturado.

Después de haber efectuado la conversión, el sistema de archivos original se encuentra en el directorio `/ext2_saved` del sistema de archivos (en realidad se trata de un subvolumen). En caso de problema, se podría recuperar el contenido.



En la versión actual del paquete `btrfs-tools` para Debian, v4.20.1, el comando está documentado pero no se encuentra en dicho paquete. A pesar de que forma parte de los elementos que se tienen que conocer para pasar la certificación, no se aconseja su uso.

c. Información del sistema de archivos Btrfs

El comando `btrfs` es una caja de herramientas que permite administrar los sistemas de archivos Btrfs. Este admite como argumento numerosos subcomandos, como `filesystem` que permite gestionar diferentes características de los sistemas de archivos Btrfs.

Sintaxis

```
btrfs sistemadearchivo sub_comando args
```

Parámetros principales

<code>sub_comando</code>	Operación que se efectuará.
<code>args</code>	Opciones de la operación que se efectuará.

Descripción

Entre los numerosos subcomandos de `btrfs sistemadearchivo`, se encuentran los que muestran elementos de información sobre las características de un subsistema, su estado y su uso:

```
show [ opciones ] [ Camino|uuid|ArchivoEspecial|label ]
```

Muestra los datos solicitados sobre el sistema de archivos Btrfs especificado por su punto

de montaje, su UUID, su archivo especial o su etiqueta, o por todos los sistemas de archivos Btrfs.

```
df [ opciones ] Camino
```

Muestra el tamaño total y el utilizado por las diferentes zonas (total, metadatos, datos) de un sistema de archivos montado en Camino.

```
usage [ opciones ] Camino
```

Muestra el uso detallado de un sistema de archivos montado en Camino.

```
du Camino
```

Muestra el tamaño de los objetos de un sistema de archivos montado en Camino.

```
label [ ArchivoEspecial|Camino ] NuevoLabel
```

Muestra o modifica la etiqueta de un sistema de archivos Btrfs.

Ejemplo

Se utilizan distintos subcomandos de `btrfs filesystem`:

```
btrfs filesystem show /dev/sdb1
```

```
Label: none uuid: 0ea1173f-9ee5-421e-ac7d-0cb374df8738
```

```
Total devices 1 FS bytes used 840.00KiB
```

```
devid 1 size 7.54GiB used 795.88MiB path /dev/sdb1
```

El sistema de archivos no tiene etiqueta, su tamaño es de 7,54 GiB de los cuales 840 KiB son utilizados. Es un monoperiférico.

btrfs filesystem df /dev/sdb1

ERROR: not a btrfs filesystem: /dev/sdb1

Hay que montar el sistema de archivos:

mount /dev/sdb1 datos1

btrfs filesystem df datos1

Data, [single](#): total=8.00MiB, used=712.00KiB

System, DUP: total=8.00MiB, used=16.00KiB

Metadata, DUP: total=385.94MiB, used=112.00KiB

GlobalReserve, [single](#): total=16.00MiB, used=0.00B

Se ven las diferentes zonas del sistema de archivos, su tamaño total y su ocupación.

btrfs filesystem de datos1

Total	Exclusive	Set shared	Filename
0.00B	0.00B	-	datos1/datos/lost+ found
108.00KiB	108.00KiB	-	datos1/datos/dir2/archivo2
0.00B	0.00B	-	datos1/datos/dir2/arch
108.00KiB	108.00KiB	-	datos1/datos/dir2
108.00KiB	108.00KiB	-	datos1/datos/dir3/archivo3
4.00KiB	4.00KiB	-	datos1/datos/dir3/arch
112.00KiB	112.00KiB	-	datos1/datos/dir3
108.00KiB	108.00KiB	-	datos1/datos/dir6/archivo6
108.00KiB	108.00KiB	-	datos1/datos/dir6
108.00KiB	108.00KiB	-	datos1/datos/dir1/archivo1
108.00KiB	108.00KiB	-	datos1/datos/dir1
108.00KiB	108.00KiB	-	datos1/datos/dir4/archivo4
0.00B	0.00B	-	datos1/datos/dir4/arch
108.00KiB	108.00KiB	-	datos1/datos/dir4
108.00KiB	108.00KiB	-	datos1/datos/dir5/archivo5
0.00B	0.00B	-	datos1/datos/dir5/arch
108.00KiB	108.00KiB	-	datos1/datos/dir5
652.00KiB	652.00KiB	-	datos1/datos
652.00KiB	652.00KiB	0.00B	datos1

El comando muestra el tamaño de cada objeto del sistema de archivos.

btrfs filesystem usage datos1

Overall:

Device size:	7.54GiB	
Device allocated:	795.88MiB	
Device unallocated:	6.76GiB	
Device missing:	0.00B	
Used:	1.05MiB	
Free (estimated):	6.77GiB	(min: 3.39GiB)
Data ratio:	1.00	
Metadata ratio:	2.00	
Global reserve:	16.00MiB	(used: 0.00B)

Data, **single**: Size:8.00MiB, Used:820.00KiB

/dev/sdb1 8.00MiB

Metadata,DUP: Size:385.94MiB, Used:112.00KiB

/dev/sdb1 771.88MiB

System,DUP: Size:8.00MiB, Used:16.00KiB

/dev/sdb1 16.00MiB

Unallocated:

/dev/sdb1 6.76GiB

*El comando muestra los volúmenes libres y utilizados por el sistema de archivos.**Se le da una etiqueta al sistema de archivos:***btrfs filesystem **label** datos1 DATOSBTRFS***Se comprueba:***btrfs filesystem **label** datos1**

DATOSBTRFS

d. Montaje de un sistema de archivos Btrfs

El montaje de un sistema de archivos de tipo Btrfs se hace como para los otros sistemas de archivos, gracias al archivo `/etc/fstab`, a una unidad de montaje `systemd` o

directamente usando el comando `mount`.

En cualquier caso, se pueden especificar muchas opciones para Btrfs, entre las cuales se encuentran:

<code>Autodefrag</code>	Desfragmentación en segundo plano.
<code>compress=zlib</code>	Compresión minimizando el espacio utilizado.
<code>compress=lzo</code>	Compresión minimizando el tiempo de compresión.
<code>compress=no</code>	Sin compresión.
<code>ssd</code>	Optimización para los discos SSD.

e. Subvolúmenes Btrfs

Los **subvolúmenes** son subconjuntos de un sistema de archivos Btrfs. Se pueden montar independientemente del sistema de archivos y se ven como directorios. Se pueden hacer instantáneas (*snapshots*) de estos subvolúmenes.

El subcomando `subvolume` del comando `btrfs` permite gestionar los subvolúmenes.

Sintaxis

```
btrfs subvolumen subcomando args
```

Parámetros principales

subcomando	Operación que se efectuará en el subvolumen.
args	Opciones de la operación que se efectuará.

Descripción

Muchos subcomandos de `btrfs subvolume` permiten administrar los subvolúmenes, entre los cuales:

```
btrfs subvolume create [Dest/]NombreSubVol
```

Crea el subvolumen `NombreSubVol` en el directorio `Dest` (por defecto, en el directorio actual).

```
btrfs subvolume list Camino
```

Muestra la lista de los subvolúmenes del sistema de archivos montado en `Camino`.

```
btrfs subvolume show Camino
```

Muestra las características del subvolumen `Camino`.

```
btrfs subvolume delete NombreSubVol
```

Suprime el subvolumen `NombreSubVol`.

Para montar un subvolumen, hay que usar el comando `mount`, especificando el sistema de archivos Btrfs que contiene el subvolumen y usando la opción de montaje `subvol=`:

```
mount ArchivoEspecialFS -t btrfs -o subvol=NombreSubVol PuntoMontaje
```

Ejemplos

Se crea el subvolumen `etc_sv` en el sistema de archivos Btrfs montado en `/root/datos1`:

```
btrfs subvolume create /root/datos1/etc_sv  
Create subvolume '/root/datos1/etc_sv'
```

Listado de los subvolumenes del sistema de archivos:

```
btrfs subvolume list /root/datos1  
ID 257 gen 16 top level 5 path etc_sv
```

El subvolumen es visto como un directorio:

```
ls -il /root/datos1  
total 0  
257 drwxr-xr-x 1 root root 68 marzo  6 18:57 datos  
256 drwxr-xr-x 1 root root  0 marzo  7 18:18 etc_sv
```

Las características del subvolumen:

```
btrfs subvolume list /root/datos1/etc_sv  
ID 257 gen 13 top level 5 path etc_sv  
btrfs subvolume show /root/datos1/etc_sv  
etc_sv  
  Name:          etc_sv  
  UUID:          6bae5bdd-f818-8f4b-962f-559187564f44  
  Parent UUID:    -  
  Received UUID:  -  
  Creation time:  2020-03-07 18:18:14 +0100  
  Subvolume ID:   257  
  Generation:     13  
  Gen at creation: 13  
  Parent ID:      5  
  Top level ID:   5
```


Flags: -
 Snapshot(s):

El subvolumen se usa como un directorio:

```
cp /etc/* /root/datos1/etc_sv 2>/dev/null
ls /root/datos1/etc_sv
adduser.conf      inputrc          passwd
adjtime           issue            passwd-
anacrontab        issue.net        profile
apg.conf          kernel-img.conf  protocols
appstream.conf    ld.so.cache      reportbug.conf
[...]
```

Se puede montar individualmente en la arborescencia global:

```
mount /dev/sdb1 -tbtrfs -osubvol=etc_sv /mnt
```

El comando `mount` muestra el montaje del subvolumen:

```
mount | grep etc_sv
/dev/sdb1 on /mnt type btrfs
(rw,relatime,space_cache,subvolid=257,subvol=/etc_sv)
```

El directorio de montaje corresponde al del subvolumen:

```
ls -lid /root/datos1/etc_sv/ /mnt
256 drwxr-xr-x 1 root root 1730 marzo  7 18:24 /mnt
256 drwxr-xr-x 1 root root 1730 marzo  7 18:24 /root/datos1/etc_sv/
```

Se puede modificar el subvolumen mediante su directorio en su sistema de archivos o a través de su punto de montaje:

```

echo "void" > /root/datos1/etc_sv/arch_sv.txt
echo "void1" > /mnt/miarch_mnt.txt
ls -l /root/datos1/etc_sv/miarch_mnt.txt /mnt/arch_sv.txt
-rw-r--r-- 1 root root 5 marzo  7 18:39 /mnt/arch_sv.txt
-rw-r--r-- 1 root root 6 marzo  7 18:39 /root/datos1/etc_sv/miarch_mnt.txt

```

Los dos archivos se encuentran en el subvolumen.

f. Instantáneas Btrfs

Una instantánea (*snapshot*) permite aislar una «imagen» del sistema de archivos Btrfs (o de un subvolumen) en el momento de la creación de la instantánea. Esta imagen se puede gestionar independientemente del sistema de archivos o del subvolumen. Al contrario que en el caso de una copia, los datos no son duplicados, el piloto Btrfs solamente almacena la diferencia entre el original y la instantánea.

Una instantánea es vista como un directorio. Se puede copiar, respaldar, y si está en modo lectura/escritura, modificar.

Esta potente funcionalidad permite mantener eficazmente diferentes versiones de un sistema de archivos (o de un subvolumen), sin consumo excesivo del espacio en el disco y con un buen rendimiento.

El subcomando `subvolume` del comando `btrfs` permite crear un snapshot.

Sintaxis

```
btrfs subvolume snapshot [ -opt ] Origen [ Nombre|Dest/Nombre ]
```

Parámetros principales

<code>-opt</code>	Opciones de creación.
<code>Origen</code>	Camino de acceso del sistema de archivos o del subvolumen origen.
<code>Nombre</code>	Nombre de la instantánea que se creará, en el directorio <code>Dest</code> .

Descripción

La opción principal es `-r`, para crear una instantánea en modo de solo lectura.

Sin los argumentos `Nombre` o `Dest/Nombre`, se creará la instantánea en el directorio actual, con el nombre de la parte del directorio (*basename*) de `Origen`.

Una vez creada, la instantánea se manipula como un directorio. Se pueden usar los comandos `cp`, `mv`, `rm`, con las opciones relativas a los directorios.

En realidad, una instantánea es un tipo particular de subvolumen Btrfs. En ese sentido, puede ser montada en la arborescencia global.

Ejemplo

Creación del snapshot `snapshot1`, del sistema de archivos Btrfs montado en `/root/datos1` en el directorio actual `/root/datos1`:

```
cd /root/datos1
btrfs subvolume snapshot /root/datos1 snapshot1
Create a snapshot of '/root/datos1' in './snapshot1'
```

El snapshot ha sido creado en el directorio actual.

```
ls -l
total 16
drwxr-xr-x 1 root root 68 marzo 6 18:57 datos
drwxr-xr-x 1 root root 1778 marzo 7 18:39 etc_sv
```

```
drwxr-xr-x 1 root root 22 marzo 7 19:16 snapshot1
```

Su contenido corresponde al del sistema de archivos original:

```
ls -l snapshot1
total 0
drwxr-xr-x 1 root root 68 marzo 6 18:57 datos
drwxr-xr-x 1 root root 0 marzo 7 19:22 etc_sv
```

Se suprime un repertorio del sistema de archivos:

```
rm -r datos
```

Todavía se encuentra en la instantánea.

```
ls snapshot1/
datos etc_sv
```

Se «restaura» el directorio:

```
cp -r snapshot1/datos /root/datos1
ls -l
total 16
drwxr-xr-x 1 root root 68 marzo 7 19:29 datos
drwxr-xr-x 1 root root 1778 marzo 7 18:39 etc_sv
drwxr-xr-x 1 root root 22 marzo 7 19:16 snapshot1
```

6. Gestión de los sistemas de archivos de tipo XFS

El administrador del sistema dispone de comandos específicos para gestionar los sistemas de archivos de tipo XFS.

a. Creación de un sistema de archivos XFS

El comando genérico `mkfs` hace una llamada específica a `mkfs.xfs` cuando se crea un sistema de archivos con la opción `-t xfs`.

Sintaxis

```
mkfs.xfs [Opciones] ArchivoEspecial
```

Parámetros principales

Opciones	Opciones de creación.
ArchivoEspecial	Camino de acceso al espacio de almacenamiento.

Descripción

Se pueden especificar muchas opciones, en particular:

`-f`

Fuerza la sobrescritura del sistema de archivos existente.

`-d size=talla`

El tamaño del sistema de archivos en número de bloques, en el caso de que no se quiera usar todo el espacio de almacenamiento.

`-l logdev=LogArchivoEspec`

Permite localizar en el espacio de almacenaje `LogArchivoEspec` el registro de transacciones.

-L **label**

Atribuye la etiqueta **label** al sistema de archivos.

Ejemplo

Creación de un sistema de archivos XFS en un volumen lógico, con posible sobrescritura de los valores por defecto.

mkfs.xfs -f /dev/vg-datos/lvol0

```
mkfs.xfs -f /dev/vg-datos/lvol0
meta-data=/dev/vg-datos/lvol0 isize=512  agcount=4, agsize=9728 blks
        =               sectsz=4096  attr=2, projid32bit=1
        =               crc=1      finobt=1, sparse=1, rmapbt=0
        =               reflink=1
data      =               bsize=4096  blocks=38912, imaxpct=25
        =               sunit=0      swidth=0 blks
naming    =version 2      bsize=4096  ascii-ci=0, ftype=1
log       =internal log   bsize=4096  blocks=1221, version=2
        =               sectsz=4096  sunit=1 blks, lazy-count=1
realtime  =none          extsz=4096  blocks=0, rtextents=0
```

b. Gestión de la etiqueta de un sistema de archivos XFS

El comando **xfs_admin** permite mostrar la etiqueta de un sistema de archivos XFS y de modificarla con la condición de que este esté montado.

Sintaxis

xfs_admin -l

Muestra la etiqueta del sistema de archivos.

xfs_admin -L etiqueta

Atribuye el label `etiqueta` al sistema de archivos (máximo 12 caracteres).

Ejemplo

```
xfs_admin -l /dev/vg-datos/lvol0
label = ""
xfs_admin -L datos_compta /dev/vg-datos/lvol0
writing all SBs
new label = "datos_compta"
```

c. Información de un sistema de archivos XFS

El comando `xfs_info` permite obtener información de un sistema de archivos XFS montado o no.

Sintaxis

```
xfs_info PuntoMontaje
```

O

```
xfs_info ArchivoEspecial
```

Ejemplo

```
xfs_info /dev/vg-datos/lvol0
meta-data=/dev/vg-datos/lvol0 isize=512 agcount=4, agsize=9728 blks
=          sectsz=4096 attr=2, projid32bit=1
=          crc=1      finobt=1, sparse=1, rmapbt=0
=          reflink=1
data      =          bsize=4096 blocks=38912, imaxpct=25
=          sunit=0    swidth=0 blks
naming    =version 2      bsize=4096 ascii-ci=0, ftype=1
log       =internal log    bsize=4096 blocks=1221, version=2
=          sectsz=4096 sunit=1 blks, lazy-count=1
```

```
realtime =none          extsz=4096  blocks=0, rtextents=0
```

d. Extensión de un sistema de archivos XFS

El comando `xfs_growfs` permite aumentar el tamaño de un sistema de archivos XFS, con la condición de que esté montado.



No se puede reducir un sistema de archivos XFS.

Sintaxis

```
xfs_growfs -D NuevaTalla PuntoMontaje
```

O

```
xfs_growfs -D NuevaTalla SistemaArch
```

Cada uno de estos comandos aumenta el sistema de archivos `SistemaArch` (montado en `PuntoMontaje`) hasta el nuevo tamaño especificado, expresado en número de bloques.

O

```
xfs_growfs -d SistemaArch
```

Este comando fuerza el nuevo tamaño al máximo del espacio disponible en el periférico.



La operación se puede efectuar incluso si el sistema de archivos está siendo utilizado.

Ejemplo

Un sistema de archivos XFS de 150 MB ha sido creado en un volumen lógico. Se aumenta este último a 200 MB, por lo tanto, se puede expandir el sistema de archivos hasta este tamaño.

```
xfs_growfs -d /dev/mapper/vg--datos-lvol0  
xfs_growfs: /dev/mapper/vg--datos-lvol0 is not a mounted XFS filesystem
```

El comando no ha funcionado porque el sistema de archivos no está montado. Se monta el sistema de archivos:

```
mount /dev/mapper/vg--datos-lvol0 /var/datos/compta  
xfs_growfs -d /var/datos/compta  
meta-data=/dev/mapper/vg--datos-lvol0 isize=512  agcount=4, agsize=9728 blks  
=          sectsz=4096 attr=2, projid32bit=1  
=          crc=1      finobt=1, sparse=1, rmapbt=0  
=          reflink=1  
data      =          bsize=4096 blocks=38912, imaxpct=25  
=          sunit=0    swidth=0 blks  
naming    =version 2      bsize=4096 ascii-ci=0, ftype=1  
log       =internal log   bsize=4096 blocks=1221, version=2  
=          sectsz=4096 sunit=1 blks, lazy-count=1  
realtime  =none          extsz=4096 blocks=0, rtextents=0  
data blocks changed from 38912 to 51200
```

El sistema de archivos ha sido agrandado hasta el tamaño máximo disponible en el volumen lógico, es decir 51200 blocs de 4096 bytes (alrededor de 200 MB).

e. Respaldo de un sistema de archivos XFS

El comando `xfsdump` permite respaldar un sistema de archivos con la condición de que esté montado.

Sintaxis

```
xfsdump [ Opciones ] -f ArchivoRespaldo ArchivoEspecial
```

Parámetros principales

<code>-f</code>	Camino de acceso hacia el archivo que contendrá el respaldo.
<code>ArchivoRespaldo</code>	
<code>ArchivoEspecial</code>	Camino de acceso hacia el sistema de archivos que se quiere respaldar.
<code>Opciones</code>	Opciones de respaldo.

Descripción

El respaldo puede ser completo, incremental o parcial. El comando trata distintos niveles de respaldo, el nivel cero indica un respaldo completo. Un respaldo puede ser interrumpido y retomado de nuevo más tarde.

Opciones principales:

`-l nivel`

Nivel de respaldo, `0` para un respaldo total (nivel por defecto).

`-s Camino`

Limita el respaldo al archivo o directorio especificado (camino relativo con respecto al

punto de montaje). La opción `-s` puede ser utilizada varias veces.

`-v`

Visualización detallada.

`-E`

Sobrescribe el archivo destino.

`-F`

Modo no interactivo.

`-R`

Vuelve a tomar una ejecución interrumpida.

Ejemplos

Respaldo completo de un sistema de archivos XFS en un archivo.

```
xfsdump -F -E -f /var/datossave.dump /dev/vg-datos/lvol0
xfsdump: using file dump (drive_simple) strategy
xfsdump: version 3.1.8 (dump format 3.0)
xfsdump: WARNING: no session label specified
xfsdump: ERROR: /dev/vg-datos/lvol0 does not identify a file system
```

El comando no ha funcionado porque el sistema de archivos no está montado.

```
mount /dev/vg-datos/lvol0 /var/datos/compta
xfsdump -F -E -f /var/datossave.dump /dev/vg-datos/lvol0
xfsdump: using file dump (drive_simple) strategy
```

```

xfsdump: version 3.1.8 (dump format 3.0)
xfsdump: WARNING: no session label specified
xfsdump: level 0 dump of pbaCOS8:/var/datos/compta
xfsdump: dump date: Wed Mar  4 17:54:00 2020
xfsdump: session id: f93a53e3-d620-479c-83e8-808c2f2611a3
xfsdump: session label: ""
xfsdump: ino map phase 1: constructing initial dump list
xfsdump: ino map phase 2: skipping (no pruning necessary)
xfsdump: ino map phase 3: skipping (only one dump stream)
xfsdump: ino map construction complete
xfsdump: estimated dump size: 26974080 bytes
xfsdump: positioned at media file 0: dump 0, stream 0
xfsdump: WARNING: erasing media
xfsdump: WARNING: no media label specified
xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 26985072 bytes
xfsdump: dump size (non-dir files) : 26960536 bytes
xfsdump: dump complete: 1 seconds elapsed
xfsdump: Dump Summary:
xfsdump:  stream 0 /var/datossave.dump OK (success)
xfsdump: Dump Status: SUCCESS

```

El comando ha hecho un respaldo en el archivo `/var/datossave.dump` de la integralidad del sistema de archivos (nivel 0 por defecto), sin hacer preguntas al operario (`-F`), con la supresión total del contenido del archivo de destino (`-E`).

f. Restauración de un sistema de archivos XFS

La restauración se hace con el comando `xfsrestore`.

Sintaxis

```
xfsrestore [ Opciones ] -f ArchivoRespaldo Dest
```

Parámetros principales

<code>-f</code>	Camino de acceso al archivo que contiene el respaldo que se quiere restaurar.
<code>ArchivoRespaldo</code>	
<code>Dest</code>	Camino de acceso al repertorio donde se quiere restaurar.
<code>Opciones</code>	Opciones de restauración.

Descripción

Este comando restaura todo o parte de un sistema de archivos respaldado con un comando.

Principales opciones:

`-v`

Visualización detallada.

`-t`

Lista el contenido del archivo de respaldo, sin restaurarlo.

`-r`

Modo acumulativo, para restaurar respaldos incrementales.

`-i`

Modo interactivo. El comando muestra una línea de comandos y espera diferentes

subcomandos.

-s Camino

Limita la restauración al archivo o al directorio `Camino`. La opción `-s` puede aparecer varias veces.

-e

No sobrescribe los archivos existentes.

-E

No sobrescribe los archivos existentes más recientes que los del respaldo.

Ejemplo

Se muestra el contenido del archivo de respaldo:

```
xfsrestore -t -f /var/datossave.dump
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 3.1.8 (dump format 3.0) - type ^C for status and control
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: pbaCOS8
xfsrestore: mount point: /var/datos/compta
xfsrestore: volume: /dev/mapper/vg--datos-lvol0
xfsrestore: session time: Wed Mar  4 17:54:00 2020
xfsrestore: level: 0
xfsrestore: session label: ""
xfsrestore: media label: ""
xfsrestore: file system id: 51b5f340-5347-437f-bc07-ccb8098cfd7
xfsrestore: session id: f93a53e3-d620-479c-83e8-808c2f2611a3
xfsrestore: media id: e529c165-4462-48ad-9df2-0ce845bbbfc5
xfsrestore: using online session inventory
```

```

xfsrestore: searching media for directory dump
xfsrestore: reading directories
xfsrestore: 4 directories and 5 entries processed
xfsrestore: directory post-processing
xfsrestore: reading non-directory files
encorso/archivo
arch/2019.tar
xfsrestore: table of contents display complete: 0 seconds elapsed
xfsrestore: Restore Summary:
xfsrestore:  stream 0 /var/datossave.dump OK (success)
xfsrestore: Restore Status: SUCCESS

```

Se restaura todo el sistema de archivos en el directorio `/var/datos/rest`:

```

xfsrestore -f /var/datossave.dump /var/datos/rest
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 3.1.8 (dump format 3.0) - type ^C for status and control
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: pbaCOS8
xfsrestore: mount point: /var/datos/compta
xfsrestore: volume: /dev/mapper/vg--datos-lvol0
xfsrestore: session time: Wed Mar  4 17:54:00 2020
xfsrestore: level: 0
xfsrestore: session label: ""
xfsrestore: media label: ""
xfsrestore: file system id: 51b5f340-5347-437f-bc07-ccb8098cfd7
xfsrestore: session id: f93a53e3-d620-479c-83e8-808c2f2611a3
xfsrestore: media id: e529c165-4462-48ad-9df2-0ce845bbbfc5
xfsrestore: using online session inventory
xfsrestore: searching media for directory dump
xfsrestore: reading directories
xfsrestore: 4 directories and 5 entries processed
xfsrestore: directory post-processing
xfsrestore: restoring non-directory files
xfsrestore: restore complete: 1 seconds elapsed
xfsrestore: Restore Summary:
xfsrestore:  stream 0 /var/datossave.dump OK (success)

```

xfsrestore: Restore Status: SUCCESS

g. Reorganizar un sistema de archivos XFS

El comando `xfs_fsr` permite reorganizar, para optimizarlo, un sistema de archivos XFS, con la condición de que este esté montado.

Sintaxis

```
xfs_fsr [ -v|d ] [ SisArch | archivo ]
```

Parámetros principales

<code>-v d</code>	Visualización detallada (<code>-v</code>) o muy detallada (<code>-d</code>).
<code>SisArch</code>	Sistema(s) de archivos.
<code>archivo</code>	Archivos (s).

Descripción

El sistema de archivos debe estar montado.

Sin argumento, el comando se aplica al conjunto de sistemas de archivos XFS montados, pero podemos limitar su acción a uno o distintos sistemas de archivos o archivos XFS.

La ejecución en el conjunto de los sistemas de archivos XFS montados puede tardar bastante tiempo.

Ejemplo

Reorganización de todos los sistemas de archivos XFS montados:


```

xfs_fsr -v
Found 4 mounted, writable, XFS filesystems
xfs_fsr -m /proc/mounts -t 7200 -f /var/tmp/.fsrlast_xfs ...
START: pass=0 ino=0 /dev/mapper/cl-root /
/ start inode=0
ino=36092035
extents before:3 after:1 DONE ino=36092035
ino=36092037
extents before:3 after:1 DONE ino=36092037
ino=69048622
ino=69048622 already fully defragmented.
/var start inode=0
ino=144
[...]
/home start inode=0
ino=134
extents before:3 after:1 DONE ino=134
ino=33554615
extents before:6 after:1 DONE ino=33554615
ino=33554566
extents before:3 after:1 DONE ino=33554566
ino=33554567
[...]

```

h. Comprobar un sistema de archivos XFS

El comando `xfs_check` comprueba un sistema de archivos XFS.

Para evitar errores en el control, el sistema de archivos debería estar desmontado o montado en modo de solo lectura.

Ejemplo

```

xfs_check /dev/sdb
xfs_check: /dev/sdb contains a mounted and writable sistema de archivos
fatal error -- couldn't initialize XFS library

```

El sistema de archivos está montado, el comando nos muestra un error.

```
umount /dev/sdb
xfs_check /dev/sdb
```

i. Reparar un sistema de archivos XFS

El comando `xfs_repair` repara un sistema de archivos XFS.

El sistema de archivos tiene que estar desmontado para poder ser reparado correctamente.

Ejemplo

```
xfs_repair /dev/sdb
Phase 1 - find and verify superblock...
Phase 2 - using internal log
    - zero log...
    - scan file system freespace and inode maps...
    - found root inode chunk
Phase 3 - for each AG...
    - scan and clear agi unlinked lists...
    - process known inodes and perform inode discovery...
    - agno = 0
    - agno = 1
    - agno = 2
    - agno = 3
    - agno = 4
    - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
    - setting up duplicate extent list...
    - check for inodes claiming duplicate blocks...
    - agno = 0
    - agno = 1
    - agno = 2
    - agno = 3
    - agno = 4
Phase 5 - rebuild AG headers and trees...
    - reset superblock...
Phase 6 - check inode connectivity...
```

- resetting **contents of** realtime **bitmap and** summary inodes
- traversing **file system** ...
- traversal finished ...
- moving disconnected inodes **to** lost+**found** ...

Phase 7 - **verify and** correct **link** counts...
done

7. Supervisión de los dispositivos SMART

El protocolo **SMART** (*Self-Monitoring, Analysis and Reporting Technology*), reconocido e implementado por la mayoría de los proveedores de dispositivos de almacenamiento, permite que una aplicación pueda interrogar al controlador de dispositivos para supervisar el estado de sus dispositivos.

a. El paquete smartmontools

Este paquete (generalmente incluido por defecto en las distribuciones) instala herramientas muy potentes de seguimiento de recursos de los discos. Implementa el servicio `smartd`, que se puede configurar para seguir el estado de los diferentes dispositivos de almacenamiento, y un comando de control, `smartctl`.

b. El servicio smartd

El servicio `smartd` está configurado para recabar cada 30 minutos la información sobre los discos de la máquina, escaneados automáticamente. Su archivo de configuración se encuentra, generalmente, en `/etc/smartmontools/smartd.conf` (y `/etc/default/smartmontools` para configurar el arranque en el caso de una distribución Debian).

Una vez iniciado, el servicio supervisa los discos que ha detectado y envía mensajes al administrador en caso de problemas.

[*Ejemplo*](#)

Mensaje generado por `smartd` para el administrador del sistema, donde se le dice que existen sectores defectuosos en un disco:

To: root@beta64.localdomain
Subject: SMART `error` (CurrentPendingSector) detected `on` host: beta64
User-Agent: Heirloom mailx 12.4 7/29/19
Content-Type: `text/plain`; charset=us-ascii
From: root@beta64.localdomain (root)
Status: R

This email was generated `by` the smartd daemon running `on`:

host name: beta64
DNS domain: [Unknown]
NIS domain: (none)

The following warning/`error` was logged `by` the smartd daemon:

Device: /dev/sda [SAT], 8 Currently unreadable (pending) sectors

c. El comando `smartctl`

Este comando da información de los discos del sistema y, particularmente, de su correcto o incorrecto funcionamiento.

Sintaxis

`smartctl [opciones] ArchivoEspecial`

Parámetros principales

<code>--scan</code>	Detecta los discos.
<code>-a</code> <code>ArchivoEspecial</code>	Información completa del disco.
<code>-i</code> <code>ArchivoEspecial</code>	Información del disco (modelo, serie, etc.).
<code>-H</code> <code>ArchivoEspecial</code>	Información del estado de fiabilidad (<i>Health</i>) del disco. Si se indica que el disco tiene un estado problemático, hay que respaldar urgentemente los datos y preparar su reemplazo.
<code>-c</code> <code>ArchivoEspecial</code>	Información sobre las capacidades de supervisión SMART del disco.
<code>-l</code> <code>ArchivoEspecial</code>	Mensajes de registro de tipo error con relación al disco.
<code>-t tipo_disco</code>	Efectúa el autotest del disco en modo <code>type</code> (<i>short, long...</i>).
<code>-l selftest disco</code>	Resultados del autotest del disco.

Ejemplo

```
smartctl --scan  
/dev/sda -d scsi # /dev/sda, SCSI device
```

Este sistema dispone de un solo disco, `/dev/sda`, de tipo SCSI.

smartctl -i /dev/sda

smartctl 6.6 2017-11-05 r4594 [x86_64-linux-4.18.0-147.5.1.el8_1.x86_64]
 (local build)
 Copyright (C) 2002-17, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===

Device Model: ST320LT020-9YG142
 Serial Number: W043X12J
 LU WWN Device Id: 5 000c50 0493da068
 Firmware Version: 0002HPM1
 User Capacity: 320 072 933 376 bytes [320 GB]
 Sector Sizes: 512 bytes logical, 4096 bytes physical
 Rotation Rate: 5400 rpm
 Device is: Not in smartctl database [for details use: -P showall]
 ATA Version is: ATA8-ACS T13/1699-D revision 4
 SATA Version is: SATA 2.6, 3.0 Gb/s
 Local Time is: Thu Mar 5 17:13:19 2020 GMT
 SMART support is: Available - device has SMART capability.
 SMART support is: Enabled

Disco SATA de 320 GB.

smartctl -H /dev/sda

smartctl 6.6 2017-11-05 r4594 [x86_64-linux-4.18.0-147.5.1.el8_1.x86_64]
 (local build)
 Copyright (C) 2002-17, Bruce Allen, Christian Franke, www.smartmontools.org
 === START OF READ SMART DATA SECTION ===
 SMART overall-health self-assessment test result: PASSED

El disco está en buen estado.

smartctl -l error /dev/sda

smartctl 6.6 2017-11-05 r4594 [x86_64-linux-4.18.0-147.5.1.el8_1.x86_64]
 (local build)
 Copyright (C) 2002-17, Bruce Allen, Christian Franke,

<http://smartmontools.sourceforge.net>

=== START OF READ SMART DATA SECTION ===

SMART Error Log Version: 1

ATA Error Count: 10 (device log contains only the most recent five errors)

CR = Command Register [HEX]

FR = Features Register [HEX]

SC = Sector Count Register [HEX]

SN = Sector Number Register [HEX]

CL = Cylinder Low Register [HEX]

CH = Cylinder High Register [HEX]

DH = Device/Head Register [HEX]

DC = Device Command Register [HEX]

ER = Error register [HEX]

ST = Status register [HEX]

Powered_Up_Time is measured from power on, and printed as

DDd+hh:mm:ss.sss where DD=days, hh=hours, mm=minutes,

SS=sec, and sss=millisec. It "wraps" after 49.710 days.

Error 10 occurred at disk power-on lifetime: 3055 hours (127 days + 7 hours)

When the command that caused the error occurred, the device was active or idle.

After command completion occurred, registers were:

ER ST SC SN CL CH DH

40 51 00 68 b4 24 e0 Error: UNC at LBA = 0x0024b468 = 2405480

Commands leading to the command that caused the error were:

CR FR SC SN CL CH DH DC Powered_Up_Time Command/Feature_Name

25 00 08 67 b4 24 e0 00 00:06:48.670 READ DMA EXT

25 00 08 5f b4 24 e0 00 00:06:48.670 READ DMA EXT

25 00 08 57 b4 24 e0 00 00:06:48.670 READ DMA EXT

25 00 08 4f b4 24 e0 00 00:06:48.670 READ DMA EXT

25 00 08 47 b4 24 e0 00 00:06:48.670 READ DMA EXT

El disco ha presentado varias veces errores de tipo **READ DMA**, probablemente debidos a sectores en mal estado.

smartctl -t short /dev/sda

smartctl 6.6 2017-11-05 r4594 [x86_64-linux-4.18.0-147.5.1.el8_1.x86_64]
 (local build)
 Copyright (C) 2002-17, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===

Sending command: "Execute SMART Short self-test routine immediately in off-line mode".

Drive command "Execute SMART Short self-test routine immediately in off-line mode" successful.

Testing has begun.

Please wait 2 minutes for test to complete.

Test will complete after Thu Mar 5 17:19:15 2020

Use smartctl -X to abort test.

El test se ejecuta en segundo plano, en modo rápido.

smartctl -l selftest /dev/sda

smartctl 6.6 2017-11-05 r4594 [x86_64-linux-4.18.0-147.5.1.el8_1.x86_64]
 (local build)
 Copyright (C) 2002-17, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF READ SMART DATA SECTION ===

SMART Self-test log structure revision number 1

Num	Test_Description	Status	Remaining	LifeTime(hours)
LBA_of_first_error				
# 1	Short offline	Completed without error	00%	5888 -

Al parecer, no hay ningún error.

Informe completo del disco.

smartctl -a /dev/sda

smartctl 6.6 2017-11-05 r4594 [x86_64-linux-4.18.0-147.5.1.el8_1.x86_64]

(local build)

Copyright (C) 2002-17, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===

Model Family: Seagate Momentus 5400.3

Device Model: ST9160821AS

Serial Number: 5MA3G4KT

Firmware Version: 3.BHD

User Capacity: 160 041 885 696 bytes [160 GB]

Sector Size: 512 bytes logical/physical

Device is: In smartctl database [for details use: -P show]

ATA Version is: 7

ATA Standard is: Exact ATA specification draft version not indicated

Local Time is: Wed Jul 16 07:58:09 2019 CEST

SMART support is: Available - device has SMART capability.

SMART support is: Enabled

[...]

ID#	ATTRIBUTE_NAME	FLAG	VALUE	WORST	THRESH	TYPE	UPDATED
-----	----------------	------	-------	-------	--------	------	---------

WHEN_FAILED RAW_VALUE

1	Raw_Read_Error_Rate	0x000f	100	253	006	Pre-fail	Always
---	---------------------	--------	-----	-----	-----	----------	--------

- 0

3	Spin_Up_Time	0x0002	100	099	000	Old_age	Always
---	--------------	--------	-----	-----	-----	---------	--------

- 0

4	Start_Stop_Count	0x0033	099	099	020	Pre-fail	Always
---	------------------	--------	-----	-----	-----	----------	--------

- 1999

5	Reallocated_Sector_Ct	0x0033	100	100	036	Pre-fail	Always
---	-----------------------	--------	-----	-----	-----	----------	--------

- 0

7	Seek_Error_Rate	0x000f	079	060	030	Pre-fail	Always
---	-----------------	--------	-----	-----	-----	----------	--------

- 13137936931

9	Power_On_Hours	0x0032	096	096	000	Old_age	Always
---	----------------	--------	-----	-----	-----	---------	--------

- 3893

10	Spin_Retry_Count	0x0013	100	100	034	Pre-fail	Always
----	------------------	--------	-----	-----	-----	----------	--------

- 0

12	Power_Cycle_Count	0x0033	099	099	020	Pre-fail	Always
----	-------------------	--------	-----	-----	-----	----------	--------

- 1855

187	Reported_Uncorrect	0x0032	092	092	000	Old_age	Always
-----	--------------------	--------	-----	-----	-----	---------	--------

- 8

189	High_Fly_Writes	0x003a	100	100	000	Old_age	Always
-----	-----------------	--------	-----	-----	-----	---------	--------

- 0

190 Airflow_Temperature_Cel 0x0022 046 040 045 Old_age Always

In_the_past 54 (0 99 55 53 0)

192 **Power-Off_Retract_Count 0x0032 100 100 000 Old_age Always**

- 732

[...]

SMART Selective **self-test log data** structure revision **number 1**

SPAN MIN_LBA MAX_LBA CURRENT_TEST_STATUS

1 0 0 Not_testing

2 0 0 Not_testing

3 0 0 Not_testing

4 0 0 Not_testing

5 0 0 Not_testing

Selective **self-test** flags (0x0):

After scanning selected spans, **do NOT read-scan remainder of** disk.

If Selective **self-test** is pending **on power-up**, **resume after 0 minute** delay.

Se puede comprobar que el disco se ha calentado de manera exagerada (**190**

Airflow_Temperature_Cel ... In_the_past ...) y que tiene sectores en error.