

# Elegir un sistema de archivos

## 1. Fundamentos

### a. Definición de sistema de archivos

La acción de “formatear” un disco, un pendrive o cualquier soporte de datos consiste únicamente en crear en un soporte de memoria secundaria (volumen de almacenamiento) la organización lógica que permite colocar datos en él. La palabra “formateo” en Linux se utiliza para describir la creación de un sistema de archivos. Hablamos de un sistema de archivos que representa a la vez la organización lógica de los soportes tanto a un nivel inferior como a un nivel de usuario.

No se escribe la información en los discos de cualquier manera. Se requiere una mínima organización para colocar en ellos tanto la información relativa a los archivos como los datos almacenados. El sistema de archivos (y los controladores asociados) es el que define esta organización. Si bien los fundamentos organizativos suelen ser los mismos en los diferentes sistemas de archivos presentes soportados por Linux, las implementaciones y organizaciones lógicas de los datos en el disco varían bastante de uno a otro. De esta manera, no hay un único tipo de sistema de archivos, sino varios, puestos a disposición del usuario, el administrador o el ingeniero.

Todos los sistemas de archivos de Linux **deben respetar las normas POSIX**. Como POSIX define un conjunto de reglas básico, un sistema de archivos puede ir más lejos de esta norma ofreciendo extensiones. La mayoría de estas conciernen a elementos de seguridad, como las ACL o selinux.

El principio básico es asociar un nombre de archivo con su contenido y autorizar su acceso: creación, modificación, supresión, desplazamiento, apertura, lectura, escritura, cierre. Conforme a este principio, el sistema de archivos debe gestionar lo que deriva de ello: mecanismos de protección de los accesos (permisos, propietarios), accesos concurrentes, etc.

### b. Representación

Además de la organización y el almacenamiento de la información y datos en los archivos,

el sistema de archivos debe facilitar al usuario una visión estructurada de sus datos, que permite distinguirlos, encontrarlos, tratarlos y trabajar con ellos en forma de archivos dentro de una estructura de directorios con los comandos asociados. Asimismo, cada sistema de archivos debe proporcionar lo necesario para que los programas puedan acceder a él.

Un sistema de archivos Unix se organiza en forma de un árbol de directorios y subdirectorios desde una raíz común. Es una estructura en árbol. Gran parte de estas se presentaron en el capítulo anterior. Cada directorio forma parte de una organización y propone, a su vez, una organización: el sistema de archivos dispone de una jerarquía ordenada. Se puede repartir la propia estructura en árbol entre varios soportes y sistemas de archivos.

### c. Los metadatos

Las **propiedades de un archivo** se denominan **metadatos**, aunque en Linux recibe el nombre de **inodo**. Los inodos o metadatos de todo archivo están ubicados en un lugar especial del soporte de almacenamiento. El contenido de los metadatos cambia de un sistema de archivos a otro. Sin embargo, en Linux, encontramos siempre este **contenido**:

- ~ los permisos;
- ~ las últimas fechas de acceso y modificación;
- ~ el propietario y el grupo;
- ~ el tamaño;
- ~ el número de bloques utilizados;
- ~ el tipo de archivos;
- ~ el contador de vínculos;
- ~ un árbol de direcciones de bloques de datos.

### d. Los nombres de los archivos

Los nombres pueden tener una longitud de **255 caracteres**. La extensión no es relevante como componente del sistema de archivos a diferencia de lo que ocurre en Windows. La extensión, que más bien podemos llamar el sufijo, permite solamente diferenciar los nombres de los archivos en función del contenido supuesto. Se determina el tipo de

archivo según su contenido, en particular el de los primeros bytes, que permiten determinar el tipo MIME. El comando **file** procede de esta manera. La extensión es una simple terminación del nombre del archivo, incluida en los 255 caracteres.

Los nombres de los archivos Unix no se establecen en los metadatos, sino en una tabla de catálogo. Gracias a ello es posible dar varios nombres a un mismo archivo.

### e. El archivo de registro

Los actuales sistemas de archivos disponen a menudo de mecanismos que permiten garantizar en la medida de lo posible la integridad de los datos. El sistema más habitual es el "journaling" (es un anglicismo). El sistema de archivos mantiene al día un archivo, el registro, en general de un tamaño dado y circular (la nueva información termina por chafar la antigua) en el cual registra todos los cambios antes de efectuarlos realmente. En caso de parada abrupta, el sistema repasa las grabaciones del registro y comprueba si se realizaron las operaciones. Si es necesario las vuelve a ejecutar. El registro contiene operaciones atómicas (n operaciones indivisibles) y, por lo tanto, incluso si éste no está completo, se asegura la coherencia de los datos o bien completando el registro cuando es posible, o bien dando un paso atrás. Por lo tanto, la reparación es mucho más fiable y rápida que en un sistema de archivos tradicional.

## 2. Los sistemas de archivos en Linux

### a. ext2

Se considera que el *second extended filesystem* ext2 es el sistema de archivos histórico de Linux, aunque éste utilizaba el MinixFS inicialmente. La primera versión, llamada ext (extended filesystem), aunque corregía los defectos de minix, tenía unos límites que impedían que fuera un verdadero sistema de archivos Unix. Ext2 es, por lo tanto, el primer sistema de archivos desarrollado específicamente para Linux, de un nivel de producción y conforme a las normas Unix (se habla de nivel de producción para indicar que un sistema cualquiera responde a los criterios de puesta en producción -uso real- en una empresa). Previsto desde el principio para soportar el aumento de funcionalidades, se sigue utilizando y mejorando desde 1993, ya que ext3 y ext4 derivan de este. Ext2 no es transaccional, lo que le confiere una desventaja durante un crash, donde la totalidad del

sistema de archivos debe ser verificada. Es lento y, a veces, peligroso.

Aunque disponga de dos sucesores (ext3 y ext4), se sigue utilizando en los soportes de tipo flash o ssd, ya que la ausencia de registro por diario implica menos escrituras y, por lo tanto, menos desgaste.

Los **archivos** pueden tener un tamaño de hasta **2 TB** (2048 GB), mientras que una **partición** puede alcanzar **32 TB, incluso 128 TB**, según el tamaño de los bloques y la arquitectura.

## b. ext3

El *third extended filesystem ext3* es el **sucesor** de **ext2** desde 1999. Cuenta con un registro. Sobre todo, es totalmente compatible con ext2. El archivo de registro es una extensión de ext2. Es posible utilizar un sistema de archivos ext3 como ext2, con los mismos comandos, las mismas operaciones. Es posible **transformar** en unos segundos un sistema ext2 en ext3, y **viceversa**. Este fue el sistema de archivos más utilizado durante cerca de 10 años. Y se encuentra todavía soportado y mantenido en muchas versiones de Linux empresarial.

Como para ext2, el tamaño máximo de los archivos es de **2 TB**, y el de una partición, de **32 TB**, siguiendo las mismas restricciones.

## c. ext4

El conocido como *fourth extended filesystem ext4* es el **sucesor** de **ext3**. Estable desde 2008, es el sistema de archivos predeterminado de un gran número de distribuciones desde 2010. Durante su fase de puesta a punto se le conocía como ext4dev. Es compatible con ext3 y, aunque es avanzado, es considerado como el sistema de archivos de transición ante la llegada de **BTRFS**. Como ext3, tiene diario. Añade nuevas funcionalidades, especialmente el concepto de extents, una técnica de preasignación de bloques en zonas contiguas que permite **limitar la fragmentación de archivos** y del disco y, por tanto, **mejora el rendimiento**. La rapidez de las comprobaciones del sistema de archivos ha sido revisada y ahora es incluso más rápido que ext3.

Un sistema de archivos ext3 puede montarse como ext4. Sin embargo, una partición ext4 sólo puede montarse como ext3 (o ext2) si el sistema de archivos ha sido creado sin el uso de **extents**.

El tamaño máximo de los archivos es de 16 TB y el de una partición de 1024 PB (Petabytes), es decir 1 EB (1 Exabyte).

#### d. BTRFS

El sistema de archivos BTRFS, que se pronuncia "ButterFS" o "BetterFS", es un sistema de archivos considerado durante mucho tiempo experimental con algunas de sus funcionalidades que todavía no son estables. Podrá convertirse en el nuevo sistema de archivos predeterminado de Linux aunque muchos editores prefieren ext4 o XFS. Algunos lo usan ya habitualmente incluyendo sistemas empleados en producción, en particular en las distribuciones SLES que lo ofrecen como el sistema de archivos por defecto. Utiliza el concepto de extent como ext4 y el de subvolumenes: la creación de árboles separados en el interior de un sistema de archivos, con opciones propias en cada árbol (permisos, cuotas, etc.). Este método también permite la creación de instantáneas (snapshots) de un árbol, la imagen exacta del árbol en un instante determinado, permitiendo comprobar las modificaciones o volver al estado inicial, como con LVM. BTRFS establece un sistema de protección de datos y de metadatos, especialmente con checksums que permiten averiguar si un archivo se ha corrompido.

Su flexibilidad le permite añadir y eliminar volúmenes y por ende redimensionar el sistema de archivos en caliente, soportando métodos RAID (0,1,5). Este es un punto fuerte en comparación con los sistemas de archivos ext.

Un sistema de archivos BTRFS se puede organizar en varios volúmenes (discos o particiones), lo que le hace parecerse a soluciones del tipo LVM.

BTRFS representa la tecnología punta como sistema de archivos para equipos de cliente y servidores.

#### e. XFS

XFS es el más antiguo de los sistemas de archivos transaccionales en Linux, y data de 1993. Creado por Silicon Graphics (sgi), y convertido a licencia libre, fue llevado a Linux en 2000 y es hoy en día propiedad de Red Hat, el cual lo ha hecho su sistema de archivos por defecto (RHEL, CentOS, Fedora). Además de sus capacidades de almacenamiento aún impensables a día de hoy, tiene un sistema de registro transaccional muy eficaz y mecanismos avanzados, como la desfragmentación en línea (en caliente y al mismo

tiempo que la escritura), la capacidad de fijar el estado de un sistema de archivos en un instante  $t$  para efectuar un backup, el dimensionamiento en caliente, la reserva de ancho de banda habilitada en comunicaciones para las entradas y salidas, una gestión avanzada de las cuotas y en particular de los directorios, etc.

El tamaño máximo (teórico, porque aún hoy nadie ha creado ninguno así de grande) de los archivos es de **8 EB (Exabytes)**. 1 EB equivale a 1024 PB (Petabyte), por lo tanto 1048576 TB, o sea, convertido a una unidad más comprensible, unos 100.000 millones de DVD. **La partición puede alcanzar 16 EB**, o sea, la capacidad máxima de un controlador en 64 bits. En sistemas de 32 bits, los tamaños se "limitan" a 16 TB.

**El uso de XFS se extiende cada vez más en Linux.**

## f. VFAT (FAT32)

**VFAT** (*Virtual File Allocation Table*) es un término genérico que agrupa las diferentes versiones de FAT que soportan los nombres de archivos largos (255 caracteres) en Windows. Se conservan estos sistemas de archivos y se siguen utilizando por razones a la vez históricas y prácticas. Durante años, la mayoría de los soportes removibles, discos externos, pendrives y lectores MP3 utilizan un sistema de archivos de este tipo. VFAT está condenado a desaparecer en algunos años.

Debido a que:

- ˘ Se trata de un sistema de archivos adaptado a los **pequeños volúmenes**.
- ˘ Es un sistema de archivos simple de implementar, ideal para lectores multimedia.
- ˘ Compatibilidad entre diversas plataformas (Windows, Linux, BSD, Mac OS, etc.).

**VFAT** padece defectos inherentes a su concepción:

- ˘ Se almacena el conjunto de la información dentro de una tabla única, incluyendo el nombre del archivo y cada dirección y longitud de los bloques (llamados clústeres) que componen los datos del archivo.
- ˘ FAT al intentar agrupar los datos de un archivo en el mayor número de clústers contiguos del soporte, provoca que el sistema se vea fuertemente fragmentado.
- ˘ Cuanto más grande es el soporte, más lento va FAT, ya que debe comprobar toda

la tabla FAT para encontrar clústeres disponibles.

- La gestión de nombres largos es puro bricolaje para muchas personas, ya que FAT continúa asegurando la compatibilidad (aún hoy en día) con los nombres cortos en 8.3.
- FAT no gestiona ningún atributo extendido, en particular ninguna noción de los permisos y propietarios.
- Un archivo FAT está limitado a 4 GB.

Aunque VFAT sea todavía el sistema de archivos predeterminado de muchos lectores multimedia, smartphones, pendrives, tarjetas de memoria y dispositivos de fotografía digital, cada vez pierde más protagonismo en relación a exFAT en el caso de los soportes de intercambio y ext3/ext4 (por ejemplo en Android).

### g. exFAT

exFAT es un sistema de archivos concebido por Microsoft especialmente en los soportes de tipo flash: pendrive, tarjetas SD, etc. Se trata del sistema de archivos utilizado por defecto en la mayoría de las tarjetas SD y MicroSD comercializadas desde hace unos años. Representan numerosas ventajas con respecto a FAT como el soporte del tamaño de los archivos y de los discos hasta 128 petabytes.

Windows, MacOS y un gran número de periféricos móviles soportan exFAT. Linux lo gestiona con FUSE (Filesystem in UserSpace). El sistema exFAT es propiedad de Microsoft, por lo tanto, sus especificaciones no se pueden conocer y hace falta usar ingeniería inversa. Sin embargo, en agosto de 2019 Microsoft publicó el conjunto de especificaciones y transfirió la patente al Open Invention Network, haciendo que exFAT se convirtiera en software libre y abriendo la vía a la implementación bajo licencia libre en el seno del kernel.

Las especificaciones de exFAT son libres, y están soportadas en el kernel 5.4. Un controlador más potente, resultado de los desarrollos de Samsung para Android, estará presente a partir de la versión 5.6 del kernel.

### h. FUSE

FUSE, Filesystem in UserSpace, es un programa que permite acceder a un sistema de

archivos en el área de usuario sin privilegios. Una biblioteca llamada libfuse se conecta con un módulo del kernel. La biblioteca y los programas que lo utilizan funcionan en el área de usuario. Si el kernel no ofrece un soporte nativo para un sistema de archivos, un programa adaptado, vinculado a la biblioteca libfuse, puede permitirle su uso. Su uso conlleva ventajas e inconvenientes:

- ˘ Se soportan numerosos sistemas de archivos físicos o virtuales: por ejemplo **ntfs**, aunque también **s3fs** (para acceder a un contenedor de objetos (**bucket**) AWS S3 como sistema de archivos), **mtpfs** (lectores multimedia, Android), **sshfs** (sftp como sistema de archivos), **wikipediafs** (navegar y modificar los artículos como un fs), **gmailfs** (cuenta gmail como sistema de almacenado), **exFAT**, etc.
- ˘ Herramientas en el área de usuario.
- ˘ Sin embargo, el rendimiento está degradado por el área de usuario y la superposición de capas.

FUSE existe para **Linux**, pero también para **macOS** y la familia de los **BSD**.