

Estructura del sistema de archivos

Un sistema de archivos son los métodos y estructuras de datos que un sistema operativo utiliza para seguir la pista de los archivos de un disco o partición; es decir, es la manera en la que se organizan los archivos en el disco.

Una estructura de sistema de archivos de un sistema operativo es el nivel más básico de organización. Casi siempre un sistema operativo interactúa con sus usuarios, aplicaciones y modelos de seguridad que dependen de la manera en que almacena los archivos en un dispositivo de almacenamiento primario (normalmente una unidad de disco duro).

Categorías

Un sistema de archivos se podría resumir en términos de dos categorías diferentes de archivos:

- Archivos compartibles vs. no compartibles
- Archivos variables vs. Estáticos

Los archivos compartibles son aquellos a los que se puede acceder desde varios hosts; mientras que los archivos no compartibles no están disponibles a todos los hosts.

Los archivos variables pueden cambiar en cualquier momento sin una intervención del gestor de sistemas; los archivos estáticos, tales como documentación, librerías y binarios, no cambian sin una actuación por parte del administrador de sistemas o de una agente que el administrador de sistemas haya escogido para realizar esta tarea.

A continuación se muestra una tabla con algunos ejemplos de estas categorías de archivos:

	Compartibles	No compartibles
Estáticos	/usr /opt	/etc /boot
Variables	/var/mail /var/spool/news	/var/run /var/lock

El hecho de que estos archivos sean vistos de esta manera es para ayudar a correlacionar la función del archivo con los permisos otorgados a los directorios que los sostienen. El modo en que el sistema operativo y sus usuarios interactúan con un archivo dado determina el directorio en el cual estos archivos están ubicados, si ese directorio está montado como de sólo lectura o sólo escritura y el nivel de acceso que cada usuario tiene a ese archivo.

El nivel superior de esta organización es crucial, como el acceso a los directorios inferiores pueden estar restringidos o se pueden manifestar problemas de seguridad si el nivel superior es dejado sin organizar o sin una estructura ampliamente usada.

El estándar FHS

En cuestión de no muchos años, Linux pasó de ser un proyecto estrictamente personal de Linus Torvalds hasta convertirse en un esfuerzo mundial que incluye a miles de colaboradores y a formar un sistema operativo robusto junto con las utilidades del proyecto GNU. Con este crecimiento se hizo necesaria la estandarización del sistema de archivos en Linux. De esta manera nació lo que hoy conocemos como el Estándar de Jerarquía del Sistema de Archivos (más comúnmente llamado FHS, del inglés Filesystem Hierarchy Standard), el cual es un documento que detalla el propósito de los directorios en un sistema Linux moderno.

Este estándar consiste en un conjunto de requerimientos y guías para la ubicación de archivos y directorios en sistemas operativos tipo-UNIX. Las guías provistas por FHS buscan tanto garantizar la interoperabilidad de las aplicaciones, herramientas de administración, herramientas de desarrollo y scripts como dar una mayor uniformidad a la documentación de estos sistemas. El estándar completo está disponible en:

<http://www.pathname.com/fhs>

El sistema de archivos raíz

El sistema de archivos raíz se simboliza con un slash (/) y como su nombre lo indica, es la raíz de la cual penden todos los otros sistemas de archivos o directorios.

Este sistema de archivos debe contener todo lo necesario para el arranque, la restauración y/o la reparación del sistema. El estándar FHS establece que los siguientes directorios o enlaces simbólicos a directorios deben estar presentes en /:

/	<i>El directorio raíz</i>
- bin/	Comandos binarios esenciales.
- boot/	Archivos estáticos del gestor de arranque - dev/ Archivos de dispositivos
- etc/	Configuración del sistema específica del host
- home/	Directorios de datos de usuarios (opcional)
- lib/	Librerías compartidas esenciales y módulos del kernel
- mnt/	Puntos de montaje para sistemas de archivos temporales
- opt/	Paquetes de software de aplicación agregados
- root/	Directorio de usuario para root (opcional)
- sbin/	Binarios esenciales del sistema
- tmp/	Archivos temporales
- usr/	Jerarquía secundaria
- var/	Datos variables

/bin: Comandos esenciales de usuarios

Es un directorio estático donde se almacenan todos los binarios necesarios para garantizar las funciones básicas a nivel de usuario. Los comandos contenidos en /bin son aquellos necesarios cuando no hay montado otro sistema de archivos.

/boot: Archivos estáticos del gestor de arranque

Este directorio contiene todo lo requerido en el proceso de arranque con excepción de los archivos de configuración y el instalador del mapa de arranque. Esto puede incluir MBR's grabados, archivos de mapa de sectores y otros datos que no son editados directamente a mano.

/dev: Archivos de dispositivos

Este directorio incluye todos los dispositivos de almacenamiento, en forma de archivos, conectados al sistema, es decir, cualquier disco duro conectado, partición, memoria USB, o CD-ROM conectado al sistema y que el sistema pueda entender como un volumen lógico de almacenamiento.

Siendo esto así, verás que la ruta en la que se encuentra cualquier volumen (partición o dispositivo externo) conectado al sistema siempre empieza por /dev.

Este es el directorio que contiene, por decirlo de algún modo, la información de cada uno de los volúmenes, a diferencia del directorio /media, que verás más adelante, que lo que contiene son solo los puntos de montaje, pero no la información real de estos volúmenes.

/etc: Configuración del sistema específica del host

Contiene los archivos y directorios de configuración del sistema.

/home: Directorios de datos de los usuarios(opcional)

Existe un subdirectorio dentro de /home por cada usuario del sistema para que cada usuario pueda almacenar sus datos personales.

/lib: Librerías compartidas esenciales y módulos del kernel

Incluye las bibliotecas esenciales que son necesarios para que se puedan ejecutar correctamente todos los binarios que se encuentran en los directorios /bin y /sbin, así como los módulos del propio kernel.

/mnt: Punto de montaje para sistemas de archivos temporales

El directorio /mnt tiene la finalidad de disponer los puntos de montaje de los distintos dispositivos de almacenamiento como por ejemplo discos duros externos, particiones de unidades externas, etc. Los medios montados en esta carpeta pueden ser tanto estáticos como variables y por norma general son compartibles.

/opt: Paquetes de software de aplicación agregados

/opt está reservado para que se puedan instalar paquetes de aplicación de terceros. Un paquete instalado en /opt debe ubicarse en un directorio separado de la forma /opt/<paquete> donde paquete es el nombre que describe al paquete de software.

/root: Directorio de usuario para root (opcional)

Esta es la ubicación recomendada para el directorio de datos de la cuenta root del sistema.

/sbin: Binarios esenciales del sistema

Aquí se almacenan los comandos y utilidades para administración del sistema, típicamente utilizados por el usuario root. Generalmente los ejecutables en /sbin sólo se usan para arrancar y montar /usr y ejecutar operaciones de recuperación del sistema.

/tmp: Archivos temporales

El directorio /tmp debe estar disponible para los programas que requieran archivos temporales.

/usr: Una jerarquía de directorios secundaria

Este es uno de los directorios más importantes del sistema puesto que contiene los programas de uso común para todos los usuarios. El directorio /usr habitualmente tiene su propia partición y debería ser montable en sólo lectura. Como mínimo, los siguientes directorios deberían ser subdirectorios de /usr:

/usr

- | - bin/ La mayoría de los comandos de usuario
- | - games/ Juegos y binarios educativos
- | - include/ Archivos de cabecera incluidos por programas de C
- | - lib/ Librerías
- | - local/ Jerarquía local
- | - sbin/ Binarios del sistema no vitales
- | - share/ Datos independientes de la arquitectura
- | - src/ Código fuente
- | - doc/ Documentación general del sistema
- | - man/ Manuales accesibles con el comando man
- | - etc/ Archivos generales de configuración
- | - X11R6/ Contiene los archivos para ejecutar Xwindow

/usr/local: Jerarquía local

La jerarquía `/usr/local` es para uso del administrador del sistema al instalar software localmente. Es similar en estructura al `/usr/`. Tiene los siguientes subdirectorios, que son similares en propósito a los del directorio `/usr/`:

```
/usr/local
|- bin/
|- doc/
|- etc/
|- games/
|- include/
|- lib/
|- libexec/
|- sbin/
|- share/
|- src/
```

/var: Jerarquía de datos variables

El directorio `/var` contiene archivos de datos variables. Esto incluye directorios y archivos de spool, datos administrativos y de logs y archivos temporales. Según lo establecido por el FHS, los siguientes directorios deberían ser subdirectorios de `/var/`:

```
/var
|- account/ Logs de process accounting (opcional)
|- cache/ Datos de cache de aplicaciones
|- crash/ Dumps de crashes del sistema (opcional)
|- games/ Datos variables de juegos (opcional)
|- lib/ Información de estado variable
|- local/ Datos variables para /usr/local
|- lock/ Archivos de bloqueo
|- log/ Archivos de log
|- mail/ Archivos mailbox de usuarios (opcional)
|- opt/ Datos variables para /opt
|- run/ Datos relevantes a procesos en ejecución
|- spool/ Datos de spool de aplicaciones
|- tmp/ Datos temporales preservados si se reinicia el sistema
```

Implementación de un Sistema de Archivos

Antes de que una partición o disco sea utilizada como un sistema de archivos, necesita ser iniciada, y las estructura de datos necesitan escribirse al disco. Este proceso se denomina construir un sistema de archivos. La mayoría de los sistema de archivos UNIX tienen una estructura general parecida, aunque los detalles exactos pueden variar un poco.

Los conceptos centrales son:

- Superbloque
- Nodo-i
- Bloque de datos
- Entrada de directorio
- Bloque indirecto

Un superbloque tiene información del sistema de archivos en conjunto, como su tamaño (la información precisa aquí depende del sistema de archivos).

Un nodo-i tiene toda la información de un archivo, salvo su nombre. El nombre se almacena en el directorio, junto con el número de nodo-i.

Una entrada de directorio consiste en un nombre de archivo y el número de nodo-i que representa al archivo. El nodo-i contiene los números de varios bloques de datos, que se utilizan para almacenar los datos en el archivo. Sólo hay espacio para unos pocos números de bloques de datos en el nodo-i; en cualquier caso, si se necesitan más, más espacio para punteros a los bloques de datos son colocados de forma dinámica.

Estos bloques colocados dinámicamente son bloques indirectos; el nombre indica que para encontrar el bloque de datos, primero hay que encontrar su número en un bloque indirecto.

Sistemas de archivos soportados por Linux

Los controladores para los principales sistemas de archivos están incluidos en el Kernel. Esto quiere decir que, salvo que los desarrolladores los hayan eliminado a propósito, cualquier distro debería reconocer los principales sistemas de archivos de Linux sin problemas. Ahora bien, no todos los sistemas están pensados para lo mismo.

Los principales sistemas de archivos que podemos usar en Linux son:

EXT4

Extended4, Fourth Extended Filesystem, o más conocido como EXT4, es el sistema de archivos usado por la mayoría de las distribuciones. A grandes rasgos, viene a ser el NTFS de Linux. Este sistema de archivos llegó para suceder a EXT3, incluyendo una gran cantidad de funciones y características, entre otras, el soporte para unidades sólidas SSD.

Entre sus características podemos destacar mejor rendimiento y fiabilidad de los datos que su predecesor, soporte para journaling y medidas de seguridad para evitar la pérdida de los datos en caso de un corte de energía. Además, gracias a las funciones de extensión y asignación retrasada, se mejora el rendimiento y se reduce la fragmentación de las unidades.

XFS

Este formato está especialmente diseñado para sistemas que realizan muchas lecturas y escrituras de datos en los discos. Ofrece un rendimiento sobresaliente incluso en situaciones de máxima carga de trabajo, y cuenta con sistemas de validación de datos para evitar perder la información guardada en las unidades. Además, gracias a sus funciones avanzadas, como los inodos asignados dinámicamente y los algoritmos avanzados, y los grupos de almacenamiento que nos permite unir unidades para sumar su espacio, se consigue un rendimiento más que excelente.

También viene incluido en el Kernel y no necesita configuración, aunque puede ser complicado instalar una distro en XFS al no ser una opción recomendada por defecto.

F2FS

Este sistema de archivos fue creado originalmente para trabajar con unidades basadas en NAND, como memorias USB o, sobre todo, unidades SSD. Este sistema de archivos fue diseñado por Samsung, aunque se ha ganado la confianza de la comunidad dado su excelente rendimiento. Cuando formateamos una unidad con él, se divide el espacio en partes muy pequeñas de manera que, en lugar de reutilizar un mismo sector una y otra vez, los datos se vayan guardando en distintas partes, alargando la vida útil de las unidades.

BtrFS

Acrónimo de «b-tree file system», BtrFS fue diseñado por Oracle con la intención de suceder a EXT. Sin embargo, aún no lo ha conseguido. Este sistema de archivos cuenta con una gran cantidad de funciones avanzadas que mejoran el funcionamiento general de todo tipo de unidades, como desfragmentación avanzada y compresión de datos. Además, permite activar las copias sombra de los datos de manera que se

mejora la persistencia y la seguridad de estos. Incluso tiene soporte para snapshots, pudiendo replicar datos, migrarlos a otras unidades o crear copias de seguridad incrementales de manera muy sencilla y eficaz.

Es compatible con RAID, aunque no está especialmente pesado para ello, sobre todo para configuraciones complejas.

OpenZFS

OpenZFS es un sistema de archivos especialmente diseñado para funcionar en sistemas RAID. Además de ser compatible con todas las configuraciones de RAID, este sistema de archivos destaca por soportar también RAIDZ, una configuración que mejora la redundancia y reduce la pérdida de datos ante un corte inesperado de energía.