There are no translations available.
En este artículo se describen algunos de los sistema de ficheros que se utilizan en los sistema GNU/Linux, así como algunas utilidades que permiten crearlos.

Sistema de ficheros GNU/Linux

Introducción.

Se puede definir el sistema de ficheros de un sistema operativo como aquellas estructuras lógicas y sus correspondientes métodos que utiliza el propio sistema para organizar los ficheros en disco.

Por contextualizar esta definición , no está de más recordar que un disco duro (soporte físico) puede poseer una o más particiones (sistema de particionado), y cada una de esas partiones requieren de formato lógico mediante un sistema de ficheros específico (durante el presente escrito se presentarán distintos tipos). Estos sistema de ficheros permiten estructurar la información para poder mostrarla, tanto de forma gráfica (si el sistema cuenta con esta característica) como de forma textual mediante los denominados gestores de archivos.

El presente artículo describe algunos de los sistema de ficheros que se utilizan de forma habitual en los sistema GNU/Linux, así como algunas utilidades que permiten crearlos.

Estructura interna de los sistemas de ficheros GNU/Linux

Para entender cómo trabaja el sistema de ficheros en los sistemas GNU/Linux, debemos tener presente que, como último objetivo, el sistema de ficheros debe permitir acceder de forma conocida a la información almacenada en la partición (normalmente, el manejador de esa información es el gestor de archivos referenciado anteriormente). Para ello, los sistema de ficheros GNU/Linux poseen una estructuración jerárquica o en árbol. Es decir, el sistema contiene unos directorios (que a su vez podrían contener más subdirectorios), que asocian características de ficheros con los ficheros guardados en la partición.

La relación descrita entre los ficheros y la forma de localizarlos se realiza, en los sistemas GNU/Linux, mediante una tabla de asignación de inodo. Un inodo contiene los parámetros

característicos del objeto referenciado (permisos, fechas, ubicación...). Este objeto puede ser tanto un fichero, un directorio, un enlace simbólico y, por generalizar, cualquier objeto que puede ser entendido por el sistema de ficheros.

Cada inodo está identificado por un número entero único (en el sistema de ficheros), y los directorios son los responsables de guardar ternas de número de inodo y nombre identificativo de fichero.

Obsérvese que el inodo no guarda el "nombre del fichero", no forma parte de su estructura. Así, cada fichero posee un único inodo, pero puede tener varios nombres que lo identifiquen.

Con esta estructura de sistema de ficheros planteada, y desde el punto de vista del usuario (ya conocemos la estructura tecnológica mediante inodos), para poder referenciar un fichero se puede utilizar la cadena de texto denominada "ruta". La ruta es el resultado de la concatenación de los nombres identificativos de directorios y subdirectorios que dibujan la estructura arbórea hasta llegar al fichero, más el nombre del propio fichero. Además, la sintaxis que sigue es estricta: la ruta empezará por el directorio superior, añadiendo a continuación los subdirectorios y por último al final el nombre del fichero; y todo ello dividido por caracteres especiales (habitualmente barras "/").

Para entenderlo mejor, pongamos un ejemplo descriptivo: la ruta al documento "escrito.odt" del usuario "user" sería:

/home/user/escrito.odt

Características del sistema de ficheros

Después de todo lo presentado, los requerimientos esperables de un sistema de ficheros pueden ser:

- poder acceder a la información (ficheros) de forma óptima
- soportar permisos de usuario, del grupo del usuario y del resto de mundo
- soportar listas de control de acceso (denominadas ACL's)
- garantizar la coherencia de la información, así como evitar la fragmentación
- permitir enlaces (simbólicos y duros)
- poder recuperar la información después de una caída de tensión brusca (journaling)

A estos requerimientos se les pueden añadir otros como poder añadir más tipos de atributos (por ejemplo, poder añadir a un fichero siempre pero sin borrar, no modificar nunca...).

Como veremos en los siguientes puntos, los sistemas de ficheros utilizados habitualmente en GNU/Linux responden a las necesidades planteadas en la mayoría de casos.

Listado de los sistema de ficheros habituales.

Los tipos de sistemas de ficheros pueden ser clasificados mediante varios criterios (rendimiento con manejo de ficheros, soporte de errores, soporte de características añadidas -por ejemplo, soporte de ACL...-), pero uno de los más habituales es el que se presenta a continuación, según su naturaleza: de disco, de red, virtuales o con propósitos especiales.

Sistema de ficheros de disco

Se trata de los sistemas de ficheros que encontramos en los dispositivos locales de los ordenadores.

- ext2 (*second extended filesystem*)/ext3 (*third extended filesystem*): son los sistemas nativos de Linux. Garantizan la compatibilidad de versiones anteriores, de modo que futuras actualizaciones no requieran rehacer el sistema de ficheros. La diferencia más importantes entre ext2 y ext3 radica en el soporte de este último de journaling.

- ReiserFS: se trata de un sistema de ficheros con jounaling desde su nacimiento, propuesto por la empresa Namesys; es la opción por defecto en algunas distribuciones (Linspire, SuSe...). Normalmente, para ficheros de tamaño pequeño tiene mejor rendimiento que ext 2 y ext3. Actualmente la versión que está soportada por la empresa creadora es la Reiser4.
- XFS :se trata de un sistema de ficheros de 64-bits con journaling y un excelente rendimiento (sobre todo con ficheros de gran tamaño)

- JFS (*journaling filesystem*): desarrollado por IBM para servidores, se trata de un sistema de ficheros de 64 bits y journaling. Fue concebido para servidores que requiriesen alto rendimiento y de ficheros de altas prestaciones. Como característica distintiva, la asignación de inodes no es estática, sino dinámica.
- ISO9660: es el sistema de ficheros estándard para volúmnes de sólo lectura como los CD-ROM.

Sistema de ficheros en red

Este tipo de sistemas de ficheros posibilitan que ordenadores clientes, a través de una red de área local, se conecten a otro servidor y accedan a sus ficheros como si tratase de recursos locales.

- NFS (*network filesystem*): desarrollado inicialmente por Sun Microsystems, suele ser la opción por defecto para sistema de ficheros en red sobre GNU/Linux. El protocolo es independiente de la máquina, del sistema operativo y del protocolo de transporte, ya que implementa o-nC RPC. Es interesante señalar que todas las opciones son síncronas (respecto al trabajo sobre el fichero).
- CIFS (*common internet filesystem*): también conocido como SMB o Samba, la implementación más utilizada es la desarrollada por Microsoft, y es utilizado en sistemas Windows. Permite compartir ficheros e impresoras por la red, y Linux puede implementar tanto la versión de servidor como de cliente. Es decir, permite la convivencia simultánea de sistemas Windows y GNU/Linux en la misma red de área local.

Sistema de ficheros virtuales

- VFS (*virtual filesystem*): se trata de un nivel lógico superior a los sistema de ficheros presentados hasta ahora. Básicamente, se trata de un interfaz entre el núcleo (kernel) y el sistema de ficheros real. De este modo, por ejemplo las aplicaciones pueden acceder a la información sin tener que preocuparse del sistema de ficheros en la que esté guardada. Escritorios como KDE o GNOME implementan sus propios sistema de ficheros virtuales, KIO y GNOME VFS respectivamente, para lograr este objetivo.
- SysFS: es un sistema de ficheros virtual que proporciona el kernel 2.6 de Linux. Básicamente, Sysfs proporciona información de los dispositivos del sistema (hardware) y sus controladores hacia el espacio del usuario, permitiendo además configurar alguno de sus parámetros.

Sistema de ficheros especiales

- SWAP: del inglés intercambiar, es el espacio de disco duro (puede ser un fichero o una partición) que se usa para guardar el estado de procesos que no se utilizan (o no caben) en la memoria física.

- GmailFS for Linux: basado en FUSE (el mecanismo de sistema de ficheros en el espacio de usuario) y desarrollado bajo Python, y permite proveer a los usuarios del conocido sistema de correo Gmail un sistema de ficheros accesible.

Creación del sistema de ficheros.

En los sistemas GNU/Linux, los sistemas de ficheros pueden ser creados desde un terminal mediante la orden "mkfs", y los parámetros adecuados (dependiendo del sistema de ficheros a crear). La opción que crea uno u otro sistema es la "-t", a la que se debe añadir el dispositivos (la partición, por ejemplo) físico donde crear el sistema de ficheros.

Por ejemplo, si tuviésemos que crear un sistema de ficheros ext3 en la segunda partición de nuestro primer disco, la orden sería:

#mkfs -t ext3 /dev/hda2

Para conocer todas las opciones, se recomienda acudir a la información del manual, utilizando la orden:

#man mkfs

Estructura de directorios y ficheros habituales de los sistemas GNU/Linux.

Una vez conocidos los sistemas de ficheros, cómo crearlos y sabiendo que los sistemas Linux utilizan sistemas jerárquicos de directorio único, vamos a presentar cuáles son los directorios habituales en estos sistemas, basándonos en concreto en la distribución Debian GNU/Linux (aunque puede hacerse extensible en su mayor parte a cualquier sistema GNU/Linux).

En primer lugar, se debe comentar que el directorio único nombrado en el párrafo anterior se representa mediante una barra "/", y es a partir de donde "cuelgan" el resto de directorios. Los más significativos son:

/boot: contiene el núcleo (kernel) e información indispensable para el arranque del sistema.

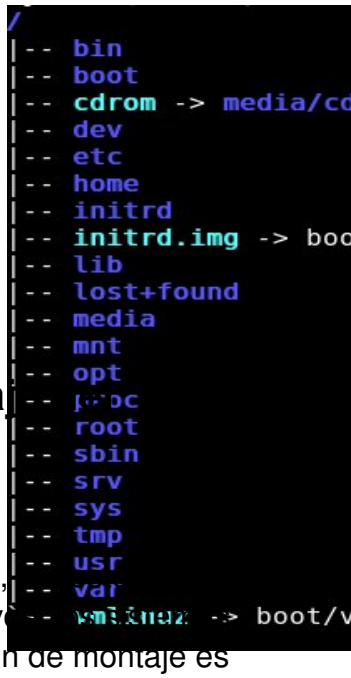
/bin: guarda unos pocos programas que estarán disponibles incluso en los modos de ejecución más restringid os (como bash, cat, ls, login, ps).

/sbin: aquí encontramos los programas disponibles sólo para el administrador incluso en los modos de ejecución más restringidos (por ejemplo fsck, getty, halt).

/usr : Programas accesibles a usuarios finales y datos de estos programas que no requieren ser modificados (datos de sólo lectura).

/floppy, /cdrom, /mnt, /media : son directorios para montar disquettes, CD-ROMs y otros sistemas de archivos o dispositivos (consultar más adelante "Montaje de sistemas de ficheros).
/lib:
Librerías indispensables y módulos (especialmente requeridas durante el arranque del sistema).
/etc:
Archivos de configuración de diversos programas y servicios.
/dev:
Abstracciones a los dispositivos conectados (o que podrían conectarse) al ordenador.
/home:
Mantiene información de los usuarios del sistema.
/root:
Mantiene información del administrador del sistema.
/tmp:
Archivos temporales creados por algunos programas, que serán borrados por el sistema operativo durante el arranque.
/var:
En este directorio los programas que lo requieran pueden mantener archivos que deban modificarse frecuentemente.
/proc:
Este directorio es virtual, no está presente en el disco, porque es creado por el sistema para intercambiar información con más facilidad.

A continuación se muestra el primer nivel del árbol de un sistema GNU/Linux:



Estructura de directorios de un sistema GNU/Linux. Primer nivel.

Monta sistema de ficheros.

Una vez creado el sistema de ficheros, para poder utilizarlo desde, por ejemplo, ficheros, se debe montar dicho sistema. A diferencia de otros sistemas operativ Linux emplean un sistema jerárquico de directorios único. Así pues, la operación de montaje es el mecanismo que integra en esa jerarquía el sistema de ficheros, de forma que sea accesible (como subdirectorio de un sistema ya montado).

Esta operación de montaje se consigue mediante la orden *mount* . Se recomienda, igual que el caso anterior, consultar las páginas *man* de esta orden.

Para entender mejor el mecanismo, vamos a plantear un ejemplo. Supongamos que tenemos nuestro sistema GNU/Linux en la primera partición del primer disco, y queremos acceder al sistema de ficheros que hemos creado anteriormente (que se encontraba en la segunda partición).

Podemos ejecutar la orden:

#mount -t ext3 /dev/hda2 /mnt

Obsérvese que la sintaxis de la orden es, además del comando, indicar el tipo de sistema de ficheros, el dispositivo físico y el directorio desde donde será accesible (punto de montaje).

De este modo, a partir de ese momento ya tenemos accesible el sistema de ficheros desde el directorio /mnt

El fichero /etc/fstab

Para no realizar manualmente cada vez que se inicia el sistema estas operaciones de montajes, se utiliza el fichero fstab, que está guardado en el directorio /etc. De este modo, cuando arranque el sistema montará aquellos sistemas de ficheros que están allí descritos.

Un ejemplo de contenido de este fichero podría ser:

 # /etc/fstab: static file system information.

#

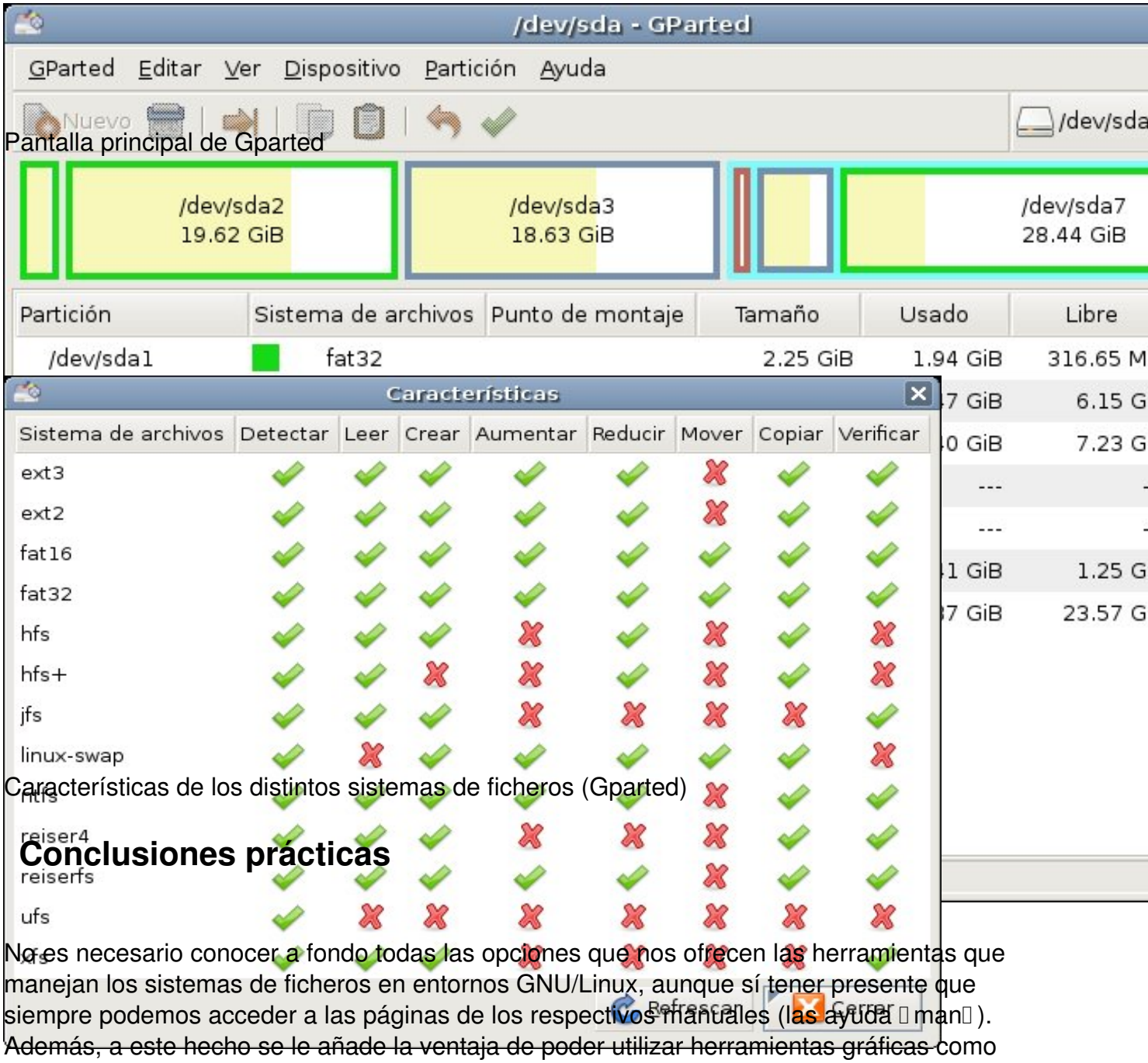
/dev/hda1 / ext3 defaults 0 1
/dev/hda2 /mnt ext3 defaults 0 0
/dev/hda3 none swap sw 0 0
none /proc proc defaults 0 0

Al instalar el sistema GNU/Linux se genera la primera versión de este fichero (según las órdenes dadas durante la instalación). Aún así, como se ha presentado, se pueden añadir nuevas entradas (por ejemplo, porque se ha añadido un nuevo disco al sistema, y se desea utilizar). Se recomienda la consulta de la información de este fichero mediante la orden: *$ man fstab*

Herramientas gráficas.

Para poder trabajar con los sistema de ficheros, así como con las particiones, existen herramientas en los entornos gráficos que facilitan esa labor. Para el escritorio KDE encontramos la utilidad QtParted, y en los escritorios GNOME la herramienta Gparted. Todas

ellas son capaces de manejar el sistema de particionado de los discos, así como dar el formato deseado a cada una de ellas. A continuación se muestran unas capturas de Gparted sobre un sistema GNU/Linux:



Pantalla principal de Gparted

Características de los distintos sistemas de ficheros (Gparted)

Conclusiones prácticas

No es necesario conocer a fondo todas las opciones que nos ofrecen las herramientas que manejan los sistemas de ficheros en entornos GNU/Linux, aunque sí tener presente que siempre podemos acceder a las páginas de los respectivos manuales (las ayuda □man□). Además, a este hecho se le añade la ventaja de poder utilizar herramientas gráficas como Gparted (incluso arrancando el sistema desde un LiveCD y □diseñando□ desde este arranque el disco físico de la máquina).

Aún así, es importante decidir previamente a cualquier actuación cuál va a ser la función del sistema, de modo que se pueda definir de forma optima tanto el sistema de particionado como, sobre todo, el sistema de ficheros a utilizar, ya que podremos obtener mejoras de rendimiento, así como v entaj as a la hora de conectar con otros sistemas de nuestra red de área local (por poner solo dos ejemplos).

Para máquinas de trabajo personales, puede ser habitual plantearse dos particiones, una con todo el sistema de ficheros en formato ext3, y una segunda partición dedicada a Swap.

No obstante, si la máquina a diseñar es un servidor, es muy posible que se pretenda una partición dedicada a compartición de ficheros, separada (en el que a particionado se refiere) del resto de la instalación, con un sistema de ficheros XFS (por su soporte de ACL), mientras que el sistema de ficheros del arranque (/boot) puede ser ext3 (para garantizar un correcto arranque de la máquina), y el resto darle formato mediante un sistema de ficheros Reiser4 (donde se nos garantiza un excelente sistema de journaling).