Estructura de directorios, archivos y puntos de montaje

Arquitectura y Sistemas Operativos.
Tecnicatura Superior en Programación.
UTN-FRA

Autores: *Prof. Martín Isusi Seff*

Revisores: *Prof. Marcos Pablo Russo*

Versión: 1

Esta obra está bajo una Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional.

Estructura de directorios

La estructura de directorios en GNU/Linux está ordenada siguiendo el estándar FHS (Estándar de jerarquía de archivos o Filesystem Hierarchy Standard), creado y mantenido por la organización Free Standards Group (Conformada por compañías de software y hardware como AMD, Debian, Dell, Google, HP, IBM, Intel, etc.). Este estándar define las bases para que tanto los programas del sitema, como los usuarios y administradores, sepan dónde encontrar lo que buscan. La mayoría de las distribuciones de GNU/Linux, inclusive las que forman parte de Free Software Standars, no aplican de forma estricta y al 100% el estándar, aunque las diferencias son mínimas. Los aspectos avanzados el estándar se pueden leer en el sitio http://www.pathname.com/fhs/pub/fhs-2.3.html.

Cuando hablamos del contenido de los directorios, podemos definir dos clasificaciones principales: estáticos/dinámicos o compartidos/restringidos

- Estáticos: Contienen binarios, bibliotecas, documentación y otros archivos que no cambian sin intervención del administrador. Pueden estar en dispositivos de sólo lectura y no es necesario que se hagan copias de seguridad frecuentemente.
- Dinámicos: Contienen archivos que no son estáticos. Deben encontrarse en dispositivos que sean de lectura-escritura. Para este tipo de directorios, sí es necesario que se realicen copias de seguridad frecuentes.
- Compartidos: Los directorios compartidos contienen archivos que pueden utilizarse en otro sistema que no sea el mismo que los almacena.
- Restringidos: Contiene archivos que no pueden compartirse.

A continuación, algunos ejemplos de directorios estáticos, dinámicos, compartidos y restringidos:

- Estáticos: **/bin, /sbin, /opt, /boot, /usr/bin**
- Dinámicos: **/var/mail, /var/spool, /var/run, /var/lock, /home**
- Compartidos: **/usr/bin, /opt**
- No compartidos: **/etc, /boot, /var/run, /var/lock**

Todos los archivos y directorios aparecen debajo del directorio raíz «/» (El equivalente en el mundo Unix al C:\ de Windows) aunque se encuentren en discos/dispositivos distintos. En Linux/Unix no existen letras de discos (C:, D:, etc.) Los dispositivos se 'montan' (empiezan a formar parte) del árbol de directorios del sistema.

A continuación. se listan los directorios más importantes y el contenido que almacenan:

Directorio	Contenido
/bin/	Comandos/programas binarios esenciales (cp, mv, ls, rm, etc.),
/boot/	Archivos utilizados durante el arranque del sistema (núcleo y discos RAM)
/dev/	Dispositivos esenciales, discos duros, terminales, sonido, video, lectores dvd/cd, etc

/etc/	Archivos de configuración utilizados en todo el sistema y que son específicos del ordenador
/etc/opt/	Archivos de configuración utilizados por programas alojados dentro de /opt/
/etc/X11/	Archivos de configuración para el sistema X Window (Opcional)
/etc/sgml/	Archivos de configuración para SGML (Opcional)
/etc/xml/	Archivos de configuración para XML (Opcional)
/home/	Directorios de inicios de los usuarios (Opcional)
/lib/	Bibliotecas compartidas esenciales para los binarios de /bin/, /sbin/ y el núcleo del sistema.
/mnt/	Sistemas de archivos montados temporalmente.
/media/	Puntos de montaje para dispositivos de medios como unidades lectoras de discos compactos.
/opt/	Paquetes de aplicaciones estáticas.
/proc/	Sistema de archivos virtual que documenta sucesos y estados del núcleo. Contiene principalmente archivos de texto.
/root/	Directorio de inicio del usuario root (super-usuario) (Opcional)
/sbin/	Comandos/programas binarios de administración de sistema.
/tmp/	Archivos temporales
/srv/	Datos específicos de sitio servidos por el sistema.
/usr/	Jerarquía secundaria para datos compartidos de solo lectura (Unix system resources). Este directorio puede ser compartido por múltiples ordenadores y no debe contener datos específicos del ordenador que los comparte.
/usr/bin/	Comandos/programas binarios.
/usr/include/	Archivos de inclusión estándar (cabeceras de cabecera utilizados para desarrollo).
/usr/lib/	Bibliotecas compartidas.
/usr/share/	Datos compartidos independientes de la arquitectura del sistema. Imágenes, archivos de texto, etc.
/usr/src/	Códigos fuente (Opcional)
/usr/X11R6/	Sistema X Window, versión 11, lanzamiento 6 (Opcional)
/usr/local/	Jerarquía terciaria para datos compartidos de solo lectura específicos del ordenador que los comparte.
/var/	Archivos variables, como son logs, bases de datos, directorio raíz de servidores HTTP y FTP, colas de correo, archivos temporales, etc.
/var/cache/	Cache da datos de aplicaciones.
/var/crash/	Depósito de información referente a caidas del sistema (Opcional)

/var/games/	Datos variables de aplicaciones para juegos (Opcional)
/var/lib/	Información de estado variable. Algunos servidores como MySQL y PostgreSQL almacenan sus bases de datos en directorios subordinados de éste.
/var/lock/	Archivos de bloqueo.
/var/log/	Archivos y directorios de registro del sistema (logs).
/var/mail/	Buzones de correo de usuarios (Opcional)
/var/opt/	Datos, variables de /opt/.
/var/spool/	Colas de datos de aplicaciones.
/var/tmp/	Archivos temporales preservados entre reinicios.

Archivos en Linux

Tanto GNU/Linux, como cualquier otro sistema operativo basado en Unix, es un sistema operativo completamente orientado a archivos. Esto quiere decir que todo (o casi todo) se representa con un archivo, tanto los datos (archivos de datos de cualquier tipo, como una canción o un programa) como los periféricos (mouse, teclado, placa de video, etc) o incluso los medios de comunicación (sockets, tuberías o *pipes*, etc.). Dado que tanto elementos de hardware, como datos, como medios de comunicación se representan como archivos, existen distintos tipos de archivos para cada caso: **los archivos directorio, los archivos comunes y los especiales.**

Archivos directorio

Los archivos directorios son una instancia especial de los archivos normales. Los directorios listan las localizaciones de otros archivos, algunos de los cuales pueden ser otros directorios.

Archivos comunes

Son archivos que pueden contener cualquier tipo de dato: Texto, Audio, Imagen, Scripts, Librerías de programación, etc.

Por defecto, nada permite diferenciar unos de otros, esto quiere decir, por ejemplo, que Linux no conoce la noción de extensión de archivo (aspecto que sí sirve para diferenciar un tipo de archivo de otro en Windows) como componente interno de la estructura del sistema de archivos, por lo que la extensión no tiene importancia y se considera simplemente parte del nombre.

Archivos especiales

Existen varios tipos de archivos especiales, pero principalmente sirven de interfaz para los diversos periféricos. Encontramos estos archivos especiales en el directorio /dev.

Nombre de archivos

Es necesario seguir reglas a la hora de nombrar archivos. Estas reglas son válidas para los tres tipos de archivos. Actualmente se puede llegar hasta 255 caracteres. Es muy importante tener en cuenta que Linux distingue entre mayúsculas y minúsculas, de modo que si tenemos un archivo nombrado "prueba", este archivo es distinto a un archivo nombrado "Prueba". Distinto sucede en, por ejemplo, en el sistema operativo Windows en cualquiera de sus versiones, donde no se diferencian mayúsculas y minúsculas y en el ejemplo antes mencionado, ambos archivos serían el mismo.

La mayoría de los caracteres (cifras, letras, ciertos signos, caracteres acentuados) son aceptados, incluyendo el espacio. A pesar de esto es necesario evitar ciertos caracteres

reservados en la shell como son : **& ; () ~ / \ ` ? –** (al principio del nombre) **<espacio>**.

Rutas de archivos

Las rutas permiten definir la ubicación de un archivo en el sistema de archivos. Es la lista de directorios y sub-directorios utilizados para acceder a un sitio determinado de la estructura hasta la posición deseada (directorio, fichero).pwd

}

El nombre de la ruta o path de un archivo es la concatenación, desde la raíz, de todos los directorios que se deben cruzar para acceder a él, que están separados cada uno por el carácter **/**.

Ejemplo: /home/usuario1/Documentos/Imagen.jpg

El ejemplo representa la ruta absoluta donde se ubica el archivo "Imagen.jpg". La ruta absoluta es aquella que se escribe desde el directorio raíz (**/**).

Además de las rutas absolutas, existen también las rutas relativas. Éstas se basan en la posición actual en el sistema de archivos. Siguiendo el ejemplo mostrado anterior, si utilizando la terminal nos ubicamos en el directorio "**usuario1**", la ruta relativa al archivo "**Imagen.jpg**" será **Documentos/Imagen.jpg**.

Para las rutas relativas se puede usar tanto el . (indica el directorio actual en el que estamos) como .. (indica el directorio superior).

Inodos y enlaces

En Linux, cada archivo en el sistema está representado por un inodo. Un inodo no es más que un bloque que almacena información de los archivos, de esta manera a cada inodo podemos asociarle un nombre. A simple vista pareciera que a un mismo archivo no podemos asociarle varios nombres, pero gracias a los enlaces esto es posible. Existen dos tipos de enlaces, los **enlaces físicos o duros** y los **enlaces simbólicos**.

Enlaces físicos o duros

Un enlace físico no es más que una etiqueta o un nuevo nombre asociado a un archivo. Es una forma de identificar el mismo contenido con diferentes nombres. Éste enlace no es una copia separada del archivo anterior sino un nombre diferente para exactamente el mismo contenido.

Para crear un enlace físico en Linux del archivo archivo.txt a nuevo_nombre.txt, ejecutamos el comando ln:

```
ln archivo.txt nuevo_nombre.txt
```

El enlace aparecerá como otro archivo más en el directorio y apuntará al mismo contenido de archivo.txt. Cualquier cambio que se haga se reflejará de la misma manera tanto para archivo.txt como para nuevo_nombre.txt.

Un enlace se puede borrar usando el comando rm de la misma manera en que se borra un

archivo, sin embargo el contenido del inodo no se eliminará mientras haya un enlace físico que le haga referencia. Esto puede tener varias ventajas, pero también puede complicar la tarea de seguimiento de los archivos. Un enlace físico no puede usarse para hacer referencia a directorios o a archivos en otros equipos.

Enlaces simbólicos

Un enlace simbólico también puede definirse como una etiqueta o un nuevo nombre asociado a un archivo pero a diferencia de los enlaces físicos, el enlace simbólico no contiene los datos del archivo, simplemente apunta al registro del sistema de archivos donde se encuentran los datos. Tiene mucha similitud a un acceso directo en Windows o un alias en OS X.

Para crear un enlace simbólico del archivo archivo.txt a nuevo_nombre.txt, ejecutamos:

ln -s archivo.txt nuevo_nombre.txt

Éste enlace también aparecerá como otro archivo más en el directorio y apuntará al mismo contenido de archivo.txt, reflejando todos los cambios que se hagan tanto para archivo.txt como para nuevo_nombre.txt.

Sobre un enlace simbólico también se pueden usar todos los comandos básicos de archivos (rm, mv, cp, etc). sin embls argo cuando el archivo original es borrado o movido a una ubicación diferente el enlace dejará de funcionar y se dice que el enlace está roto.

Un enlace simbólico permite enlazar directorios y también permite enlazar archivos fuera del equipo.

Puntos de montaje

La estructura de los sistemas de archivos está generalmente dividida en particiones, unidas todas ellas en el punto de montaje raíz (/) o separadas . Los sistemas de archivos de los dispositivos removibles como un USB o un Disco CD se unen a la raíz del sistema de la misma manera, como directorios o puntos de montaje. En principio estos directorios destinados a los dispositivos están vacíos, a la espera de su montaje, puede darse el caso de que el directorio destinado a este fin contenga subdirectorios o archivos, en cuyo caso quedarán ocultos hasta que el dispositivo se desmonte.

Para que las diferentes particiones estén disponibles desde un primer momento es necesario montarlas durante el arranque del sistema, los dispositivos removibles también se usan frecuentemente y es aconsejable tenerlos preparados para usar los comandos de montaje. Toda esta información se guarda en el archivo **/etc/fstab**. Los sistemas de archivos definidos en este fichero son revisados y montados durante el arranque del sistema. Sus entradas se consultan como fuente de información por defecto cuando los usuarios quieren montar dispositivos removibles.

Comando *mount* y *umount*

Después del arranque se pueden añadir más sistemas de archivos manualmente con el comando mount. El comando *mount* se usa para montar sistemas de archivos dentro de la estructura del árbol del sistema. El comando ***mount*** admite dos tipos de opciones, unos para el comando en sí, y otros para especificar opciones del sistema de ficheros.

mount [opciones] [dispositivo|directorio]

Las opciones que se pueden utilizar con el comando *mount* son:

-a	Lee el archivo **/etc/fstab** y monta todos los filesystem menos los que tengan la opción **noauto**
-h	Ayuda del comando
-o	Especifica las opciones del ***mount*** en la línea de comandos.
-r	Monta el filesystem en modo de solo lectura.
-t sftype	Especifica un tipo de filesystem.
-v	Salida interactiva.
-w	Monta el filesystem en modo de lectura/escritura.

Las opciones que pueden utilizarse con el comando ***mount*** y la opción ***-o*** son:

async	Toda la E/S al sistema de ficheros debería hacerse asincrónicamente.
auto	Puede montarse con la opción ***-a***
defaults	Establece las opciones: ***rw, suid, dev, exec, auto, nouser*** y **async.** (todas las opciones por defecto).
dev	Interpretar dispositivos especiales de caracteres o bloques en el sistema de archivos.
exec	Permite sólo la ejecución de binarios.
noauto	Sólo puede montarse explícitamente (esto significa que la opción ***-a*** no hará que el filesystem se monte automáticamente).
noexec	No permite la ejecución de ningún binario en el sistema de archivos montado. Esta opción puede ser útil para servidores que tienen sistemas de archivos que contienen binarios para otras arquitecturas distintas.
nouserid	No permitir el efecto de los bits bits ***SUID*** ni ***SGID***.
ro	Monta el filesystem en modo de sílo lectura.
rw	Monta el filesystem en modo de sólo escritura.
suid	Permite el efecto de los bits bits ***SUID*** ni ***SGID***.
sync	Toda la E/S al filesystem debe hacerse sincrónicamente.
user	Permite a un usuario ordinario montar el filesystem.
users	Permite a cualquier usuario el montaje/desmontaje del filesystem.
remount	Vuelve a montar un filesystem ya montado.

Empaquetado de archivos y directorios utilizando el comando *tar*

El comando ***tar*** es utilizado normalmente para empaquetar o desempaquetar archivos. La sintaxis del comando es:

tar [parámetros] [archivo1] [archivo2]

Las opciones que se pueden usar con el comando ***tar*** son:

c	Crear un archivo.
v	Muestra la salida por pantalla.
x	Extrae los archivos.
z	Comprime/descomprime **tar** utilizando **gzip** (la extensión del empaqueado será **.tar.gz** o **.tgz**).
j	Comprime/descomprime **tar** utilizando **bzip** (la extensión del empaqueado será **.tar.bz2** o **.tbz2** o **.tbz**).
f	Imprime el menú con las opciones.
t	Muestra el contenido de un paquete.