
Estructura de directorios

Diseño y Administración de Sistemas y Redes

Juan Céspedes <cespedes@gsync.es>



Curso 2005–2006



Un poco de historia

1

Un poco de historia

- Los primeros UNIX tenían una estructura de directorios muy distinta a la que tenemos ahora: directorios de usuarios en `/usr`, ejecutables de administración en `/etc...`
- En los inicios de Linux, las distribuciones situaban cada fichero donde querían.
- Era necesario estandarizarlo. Surgió el “Linux Filesystem Standard” (FSSTND).
- Más tarde, pasó a llamarse “Linux Filesystem Hierarchy Standard” (FHS).

¿Qué es un sistema de ficheros?

- Estructuras de datos que utiliza un sistema operativo para almacenar y organizar los ficheros en un disco.
- Para poder empezar a almacenar información en un sistema de ficheros, éste tiene que ser *inicializado*.
- Para poder usarlo, hay que *montarlo* en alguna parte de una estructura jerárquica de directorios, un árbol cuya raíz es el directorio llamado `/`.
- Mayúsculas y minúsculas son distintas en los nombres de fichero.

El directorio raíz

- El directorio raíz no debería contener nada que no sean directorios.
- Directorios definidos por el FHS: `bin boot dev etc home lib media mnt opt root sbin srv tmp usr var`.
- Las distribuciones y los administradores locales pueden usar más directorios. Ejemplos típicos son: `initrd lost+found proc sys`.
- Normalmente no hay nada aparte de estos directorios, con la posible excepción de imágenes del *kernel* necesarias para el arranque del sistema (`vmlinuz, initrd.img...`)

/bin: Órdenes esenciales para todos los usuarios

- Órdenes que pueden usar todos los usuarios, administradores o no, necesarias para el funcionamiento del sistema incluso cuando no hay otros sistemas de ficheros montados.
- También debe incluir la *shell* y las órdenes que ejecutan indirectamente los *scripts*
- Requisitos mínimos:

```
cat chgrp chmod chown cp date dd df dmesg echo  
false hostname kill ln login ls mkdir mknod  
more mount mv ps pwd rm rmdir sed sh stty su  
sync true umount uname
```

/boot: Ficheros requeridos para el arranque del kernel

- Los ficheros exactos dependen del sistema operativo, de la arquitectura, de la distribución y del cargador de arranque.
- En Linux tendremos, por cada *kernel* que esté configurado en el arranque:
 - **vmlinux**: Imagen del *kernel* a cargar.
 - **initrd.img**: Sistema de ficheros inicial, encargado de configurar y arrancar el sistema.
 - **config**: Fichero de configuración del *kernel*.
 - **system.map**: Lista de símbolos del *kernel* y sus direcciones en memoria.
 - **vmlinux**: Imagen del *kernel* como un ejecutable.

/dev: Dispositivos

- Todos los *dispositivos* del sistema (de caracteres y de bloques) deben estar en este directorio.
- Además de los dispositivos, suele haber un *script* encargado de crear otros nuevos, **MAKEDEV**.
- Cada dispositivo se identifica con 2 números: *major* y *minor*.
- Lista de todos los dispositivos reconocidos por el *kernel* en: `Documentation/devices.txt`.
- Algunos dispositivos no se refieren a *hardware* concreto: **mem**, **kmem**, **null**, **port**, **zero**, **full**, **random**, **urandom**...

/etc: Ficheros de configuración

- Cada subsistema, cada paquete, tiene su configuración en un fichero o un directorio bajo `/etc`
- Configuración general del sistema:
exports, **fstab**, **ftpusers**, **group**, **host.conf**, **hosts**, **inetd.conf**, **inittab**, **init.d**, **issue**, **motd**, **mtab**, **passwd**, **profile**, **protocols**, **rcN.d**, **resolv.conf**, **securetty**, **services**, **shadow**, **shells**, **skel**, **syslogd.conf**...
- Cada paquete puede tener un fichero de configuración o un directorio con varios ficheros (como `/etc/X11`).

/home: Directorios de inicio de los usuarios

- Cada usuario ha de tener un “directorio de inicio”, que es el directorio en el que empezará a ejecutarse su *shell*, donde los usuarios guardarán sus datos, etc.
- Normalmente, los directorios de inicio de cada usuario estarán en `/home/login`.
- Dentro de cada directorio de usuario habrá ficheros de configuración de varios programas, que suelen comenzar por “.”.
- Al crear una cuenta nueva, se copia en ella el contenido de `/etc/skel`.

/lib: Bibliotecas dinámicas esenciales y módulos del *kernel*

- Los ejecutables de `/bin` y `/sbin` solo pueden depender de bibliotecas dinámicas que estén en `/lib`.
- El nombre de las bibliotecas dinámicas tiene siempre el formato `/lib/libnombre.so.versión`, aunque este nombre puede ser un enlace simbólico hacia el real.
- Se puede saber de qué bibliotecas dinámicas depende un determinado ejecutable con la orden `ldd`:

```
cespedes@blas:~$ ldd /bin/cat
libc.so.6 => /lib/tls/libc.so.6 (0xb7ea8000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0xb7fea000)
```
- A partir de `/lib/modules` hay módulos para el *kernel*.

/media: Puntos de montaje de dispositivos extraíbles

- A partir de este directorio existen otros, que se utilizan como puntos de montaje adicionales a los discos duros del sistema.
- Los más habituales son:

floppy	Unidad de disquete
cdrom	Unidad de CD
usb	Dispositivo externo USB

- Suelen aparecer en el fichero `/etc/fstab`, casi siempre asociados a las opciones de montaje `user` y `noauto`.

/mnt: Punto de montaje temporal

- Normalmente es un directorio vacío.
- Se utiliza como punto de montaje temporal, a discreción del administrador, y sin que aparezca constancia alguna en el fichero de configuración `/etc/fstab`.

/opt: Aplicaciones adicionales

- Todos los ficheros que forman parte de una distribución deben estar distribuidas en distintos directorios, según el FHS.
- El *software* adicional que se instale, y que provenga de terceras partes (que no sea parte de la distribución) debe ir en */opt/paquete*.
- Dentro de cada */opt/paquete* debería haber otra jerarquía: */opt/paquete/bin*, */opt/paquete/lib*, etc.

/root: Directorio de inicio del usuario root (opcional)

- El directorio de inicio del usuario `root` suele estar aquí, en lugar de dejarlo en `/home` junto con el resto, porque el directorio `/home` puede pertenecer a un sistema de ficheros distinto del raíz, y el usuario `root` puede necesitar acceder a su directorio incluso cuando no están montados todos los discos.

/sbin: Órdenes esenciales para el administrador

- Órdenes que normalmente solo ejecutará un administrador, sea utilizando la cuenta de `root` o no, necesarias para el funcionamiento del sistema incluso cuando no hay otros sistemas de ficheros montados.
- Órdenes habituales:
`fdisk fsck getty halt ifconfig init mkfs mkswap
reboot route swapon swapoff`

/srv: Datos de servicios proporcionados por este sistema

- Lugar reservado para almacenar datos de aplicaciones o servicios que se ejecuten en esta máquina, para consulta del resto de los usuarios.
- En la práctica, normalmente no se usa.

/tmp: Ficheros temporales

- Directorio con permiso de escritura para todos los usuarios.
- Cualquier programa o *script* puede usar este directorio para dejar ficheros temporales.
- No se debe suponer que los ficheros van a preservarse de una invocación del programa a otra.
- Normalmente, el /tmp se borra en cada reinicio del sistema.
- Puede haber tareras programadas para limpiarlo periódicamente.

La jerarquía /usr

- /usr es el segundo directorio más importante del sistema, después del raíz.
- Contiene datos que se pueden compartir entre máquinas, y todo su contenido ha de ser de solo lectura.
- Cualquier información específica de una determinada máquina deberá ir en otro sitio (normalmente /etc o /var).
- Debe contener al menos los directorios **bin include lib local sbin share**.
- Opcionalmente puede haber otros, como **X11R6 games src**.

/usr/bin: Órdenes normales para todos los usuarios

- La gran mayoría de las órdenes del sistema pertenecen a **/usr/bin**.
- Cualquier paquete de *software* que no sea esencial ha de situar aquí sus órdenes, con la excepción del sistema gráfico X.

/usr/include: Ficheros *include* para preprocesado de programas en C

- Todos los ficheros *include* necesarios para programar en C deben estar en este directorio (como `stdio.h`, `stdlib.h`...)
- Los ficheros *include* del *kernel* deben estar en el directorio **/usr/include/linux**.

/usr/lib: Bibliotecas, datos estáticos de paquetes, etc

- Las librerías dinámicas que no sean usadas por ninguna orden esencial deben estar aquí.
- También tienen cabida las bibliotecas estáticas y los ficheros `.so`, necesarios para compilar programas.
- Por último, todos los datos estáticos, de solo lectura, que se pueda compartir pero en general dependiente de la arquitectura, de cualquier paquete, deben estar en `/usr/lib/paquete`.

/usr/local: Jerarquía local

- Jerarquía similar a la de `/`, `/usr` o `/opt`, pero para aplicaciones instaladas localmente por el administrador.
- Debe contener únicamente los directorios `bin` `etc` `games` `include` `lib` `man` `sbin` `share` `src`.

/usr/sbin: Órdenes normales para uso del administrador

- Órdenes de administración o de uso interno del sistema, que no es necesario que existan para reparar el sistema o en una emergencia.
- Ejemplo: órdenes para crear / borrar usuarios, servicios habituales del sistema (SSH, FTP, etc).

/usr/share: Datos independientes de la arquitectura

- /usr debe incluir datos compartibles y de solo lectura, pero probablemente dependientes de la arquitectura. /usr/share contiene datos de solo lectura e independientes de arquitectura.
- Ejemplo: documentación, ficheros de datos con formatos estándar y usables en distintos tipos de máquinas, etc.
- Puede contener los directorios `man misc dict doc games info` y algunos otros.
- Cualquier paquete puede dejar sus datos independientes de arquitectura en `/usr/share/paquete`.

/usr/X11R6: Sistema de ventanas X, Versión 11 Release 6

- Jerarquía reservada al sistema de ventanas X.
- Contiene los directorios:
 - bin** Órdenes
 - include** Ficheros *include* para compilar en C
 - lib** Bibliotecas (dinámicas y estáticas)
 - man** Páginas de manual

/usr/src: Código fuente

- El código fuente de cualquier paquete debe estar en este directorio.
- Este código fuente que debe usarse solo como referencia. **/usr/src** no es el lugar apropiado para compilar dicho código.

/var: Datos variables

- Cualquier dato que pueda sufrir modificaciones ha de ir en `/var`.
- Ejemplo: bloqueos de ficheros, correo electrónico, páginas web, información temporal de demonios, cachés de disco, *logs* del sistema, copias de seguridad, noticias, etc.
- Subdirectorios: `cache games lib local lock log mail opt run spool tmp www`.

/proc: Información del kernel

- Los directorios `/proc` y `/sys` no se corresponden con discos físicos, sino que son un medio de enviar y recibir información directamente del *kernel*.
- Cada vez que se lee o se escribe algún fichero del `/proc`, lo que se hace en realidad es pedir o enviar determinada información al *kernel*.
- En el `/proc` existe un directorio por cada PID del sistema, con información acerca del proceso con dicho PID.
- Otros ficheros tienen información general: `cmdline cpuinfo devices dma filesystems interrupts ioports loadavg meminfo modules partitions swaps uptime version...`