

Capítulo 4

Comandos básicos II

Estructura de los directorios general

Existen varias sub-jerarquías de directorios que poseen múltiples y diferentes funciones de almacenamiento y organización en todo el sistema. Se pueden clasificar en :

- **Estáticos** : Solamente root (super administrador) puede cambiar los archivos, sin embargo, pueden ser leídos por cualquier otro usuario en los directorios (/bin, /sbin, /opt, /boot, /usr/bin, etc).
- **Dinámicos** : Son archivos que son cambiantes, y pueden leerse y escribirse (algunos sólo por su respectivo usuario y el root). Entre ellos encontramos (/home, /var/mail, /var/spool, etc).
- **Compartidos** : Archivos que se pueden compartir entre usuarios pueden encontrarse en otro ordenador.
- **Restringidos** : Contiene archivos que no pueden compartir, solo son modificados por el administrador (root) (/etc, /boot, /var/run, /var/lock, etc).

Directorio	Descripción
/	Directorio principal root o raíz o contenedor de todo el sistema jerarquía.
/bin	Aplicaciones de comandos binarios los usuarios pueden ejecutar los comandos que se encuentran ahí.
/boot	Contiene el kernel (núcleo del sistema operativo) y el initrd .
/dev	Contiene los dispositivos esenciales, almacenamiento, impresoras, etc.
/etc	Archivos de configuración de nuestro sistema, nombre del host, red, de los distintos programas que instalamos, etc.
/etc/opt	Archivos de configuración de los programas instalados en /opt .
/etc/X11	Archivos de configuración para servidores X Windows System.
/etc/sgml	Archivo de configuración para SGML.
/etc/xml	Archivo de configuración para XML.
/home	Contiene los usuarios del sistema, el administrador está ubicado en el directorio /root . Por ejemplo /home/pablo el nombre de usuario es un directorio que se encuentra dentro de /home.
/lib	Están las bibliotecas esenciales compartidas de los programas alojados es decir para los binarios que se encuentran dentro de /bin y /sbin .
/media	Contiene los puntos de montajes del cdrom, usb, discos duros, etc.
/mnt	Es semejante al directorio /media .
/opt	Programas opcionales estáticas, es decir pueden ser compartidas por los usuarios.

	Guardan sus configuraciones, manuales, binarios, etc dentro de este directorio creando una estructura similar al raíz (/).
/proc	Contiene principalmente archivos de texto que contiene los procesos, información de las irq, memoria, interrupciones, filesystem, etc. Es un sistema de archivos virtuales que documenta al kernel (núcleo).
/root	Directorio del administrador (root).
/sbin	Sistemas de binarios esenciales, comandos y programas exclusivos que solo el root (super administrador) puede ejecutar. Por ejemplo init , fdisk , ifconfig , etc, algunos comandos los puede ejecutar el usuario común siempre y cuando tenga los permisos adecuados.
/srv	Lugar específico de datos que son servidos por el sistema.
/tmp	Archivos temporales, del sistema, edición, navegadores, etc.
/usr	Dentro de este directorio tendremos varias jerarquías en la cual detallaremos a continuación.
/usr/bin	Comandos binarios no administrativos para todos los usuarios.
/usr/include	Archivos de cabecera (headers files o include files) son instalados por los programas para luego poder utilizarlos en nuestro programas.
/usr/lib	Bibliotecas compartidas de los binarios.
/usr/sbin	Sistema de binarios no esenciales; por ejemplo demonios, etc. Generalmente se ejecutan al inicio del sistema o en ciertas circunstancias.
/usr/share	Aca encontraremos toda la documentación (/usr/share/doc) y manuales correspondiente a los programas instalados (/usr/share/man) e informacion (/usr/share).
/usr/src	Acá podremos encontrar el código fuente de los programas, como por ejemplo el kernel.
/usr/X11R6	Este directorio se relaciona con el X Windows System Version 11.
/usr/local	Para los datos locales del host, se crean varios subdirectorios como bin/ , lib/ , share/ , etc.
/var	Dentro de este directorio veremos varios subdirectorios que explicaremos a continuación.
/var/cache	Memoria caché de las aplicaciones, aunque también se utiliza el directorio /tmp para lo mismo.
/var/lock	Archivos Lock . Archivos que hacen el seguimiento de los recursos que se utilizan actualmente.
/var/logs	Archivos de logs del sistema y de aplicativos.
/var/mail	Buzón correos o mensajes de los usuarios. Si no utiliza cifrado, generalmente se utiliza entonces la carpeta personal para la misma labor por parte de programas que manejen correos electrónicos.
/var/opt	Posee los datos variables de /opt .
/var/run	Información reciente. Trata acerca del funcionamiento del sistema desde el último arranque. Por ejemplo, los usuarios actualmente registrados o logueados, que han ingresado; y los demonios que están en ejecución.
/var/spool	Tareas a ser procesados por ejemplo : colas de impresion, correo no leidos, etc.
/var/spool/ma	Ubicación de los correos de los usuarios.

il	
/var/tmp	Archivos temporales que, a diferencia de /tmp , no se borran entre sesiones o reinicios del sistema, pero que de todas maneras siendo prescindibles.

http://es.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

Tipos de archivos y directorios

Bajo *GNU/Linux* (y en *UNIX* en general), cualquier cosa se abstrae en forma de un archivo.

Archivos normales

Los archivos contienen datos a son ejecutables, y el sistema operativo no hace suposiciones sobre su contenido.

Directorios

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.

Enlaces duros

Cada archivo en el sistema de archivos de *GNU/Linux* tiene su propio *i-nodo*. Un *i-nodo* guarda los atributos del archivo y su posición en el disco. Si necesita referirse a un archivo usando dos nombre de archivos distintos, puede crear un *enlace duro*. El *enlace duro* tendrá el mismo *i-nodo* que el archivo original y guarda el mismo aspecto y comportamiento que el original. Con cada *enlace duro* creado, se incrementa la cuenta de referencia. Cuando se borra un *enlace duro*, se decrementa la *cuenta de referencia*. Hasta que la *cuenta de referencia* no alcanza a cero, el archivo permanece en el disco.

Note que un *enlace duro* no puede existir entre dos archivos de particiones separadas. Esto es debido a que el *enlace duro* se refiere al archivo original por *i-nodo* y el *i-nodo* de un archivo difiere entre sistemas de archivos.

Enlaces simbólicos

A diferencia de los *enlaces duros*, los cuales apuntan a un archivo por su *i-nodo*, un enlace simbólico apunta a otro archivo por su nombre. Esto permite que los enlaces simbólicos apunten a archivos localizados en otras particiones, incluso en otras unidades de red.

Enlaces de archivos: *ln*

El comando *ln* nos permite establecer *enlaces duros* y *enlaces simbólicos*.

Opciones :

-v, --verbose	<i>Verbose (muestra lo que se está borrando).</i>
-s, --symbolic	<i>Crea un enlace simbolico.</i>
-f, --force	<i>Borra los ficheros de destino que ya existieran.</i>

```
-i, --interactive   Pregunta antes de borrar ficheros de destino existente.  
-b, --backup      Hace copias de respaldo de ficheros que estén a punto de ser  
sobreescritos o borrados.  
-n, --no-  
dereference       Crea un enlace duro.
```

ln [opciones] origen destino

Ejemplo :

```
# ln -s /etc/passwd mi_passwd  
  
# ls -l mi_passwd  
  
lrwxrwxrwx 1 root root 11 abr 29 10:18 mi_passwd -> /etc/passwd  
  
# ln -n /etc/passwd mi_passwd  
  
# ls -li /etc/passwd mi_passwd  
  
149777 -rw-r--r-- 2 root root 1585 abr  4 16:50 /etc/passwd  
149777 -rw-r--r-- 2 root root 1585 abr  4 16:50 mi_passwd
```

Directorio /proc y /sys

El sistema de archivos "**proc**" se monta en el directorio **/proc** es un sistema fichero virtual, no se encuentra en ningún dispositivo físico, sino que es construido y presentado dinámicamente cada vez que le pedimos al núcleo que lo muestre, y lo mismo ocurre cuando visualizamos el contenido de sus archivos y subdirectorios. Este tipo de sistema de archivos se denomina sistema de archivos *virtual*. El contenido del directorio refleja el estado actual del núcleo de Linux, y evidentemente este estado varía con el tiempo y de un sistema a otro (por ejemplo, por disponer de hardware distinto).

Veremos que lo monta en el archivo **/etc/fstab**

```
# more /etc/fstab  
  
# <file system> <mount point> <type> <options> <dump> <pass>  
proc          /proc        proc         defaults    0           0  
...
```

Ejemplo :

```
# cat /proc/cpuinfo  
# cat /proc/swaps  
# cat /proc/version  
# cat /proc/meminfo  
# cat /proc/xxx/limits  
# cat /proc/cmdline  
# cat /proc/modules
```

Directorio /sys

El `/sys` utiliza el sistema de archivos virtual **sysfs** nuevas específicas para el kernel 2.6. Con el apoyo creciente de dispositivos **plug** caliente de hardware en el kernel 2.6, el `/sys` contiene información similar al filesystem `/proc`, pero muestra una vista de dispositivo jerárquico de la información específica a los dispositivos de conexión en caliente.

Ejemplo :

```
# ls /sys/block
# ls /sys/module
```

Comando lshal

Comando **lshal** (saca información del modelo de servidor, versión del firmware, serial del server, versión de la Bios, versión del kernel, todo tipo de información del procesador, información de los discos, puntos de montaje, particiones, así como de los usb, pci y un largo etcetera...

Opciones :

```
-m, --monitor    Información del dispositivo del monitor.
-s, --short      Imprimir sólo parte de la UDI no estáticos.
-t, --tree       Lo imprime en forma de árbol.
-u, --show <uid> Muestra solamente la especificación de un dispositivo.
-h, --help       Ayuda.
-V, --version    Version.
```

Ejemplo :

```
# lshal -m
# lshal -s
# lshal -t
# lshal -u /org/freedesktop/Hal/devices/pci_1002_5957
```

Comando lspci

Lista los dispositivos pci. Sería como un ls de pci.

Opciones :

```
-v               Verbose (-vv para mas detalle).
-k               Muestra los controladores del núcleo que maneja cada dispositivo.
-n               Muestra número de ID's.
-nn              Mostrar los servicios de identificación textual y numérica de (nombres y números).
-q               Consulta el PCI ID de base de datos para la identificación de desconocidos a través de DNS.
-qq              Como el anterior, pero re-consulta a nivel local en caché de las entradas.
-Q              Consulta la base de datos PCI ID para la identificación de todos
```

a través de DNS.

Ejemplo :

```
# lspci -v  
# lspci -vv  
# lspci -n  
# lspci -k
```

Comando lsusb

Lista los dispositivos usb. Listaría los puertos usb.

Opciones :

-v, --verbose	<i>Verbose.</i>
-s [[bus]:] [devnum]	<i>Muestra especificaciones de un bus/dev específico.</i>
-d [vendor]: [product]	<i>Muestra especificaciones un vendedor y un ID de producto.</i>
-t	<i>Lo muestre en forma de árbol.</i>

/var/lib/usbutils/usb.ids = Una lista de todos los USB ID's (proveedores, productos, clases, subclases y protocolos).

Ejemplo :

```
# lsusb  
# lsusb -s 005:001  
# lsusb -d 1d6b:0001
```

Comando lscpu

Información de la arquitectura del CPU.

Opciones :

-h, --help	<i>Imprime la ayuda.</i>
-p, --parse	<i>Imprime la salida en un formato compacto.</i>

Ejemplo :

```
# lscpu  
# lscpu -p
```

Comando arch

Muestra la arquitectura (`uname -m`).

Ejemplo :

```
# arch
# uname -m
```

Comando uname

Muestra información del sistema.

Opciones :

```
-a, --al          Muestra toda la información.
-s, --kernel-name Imprime el nombre del kernel.
-r, --kernel-    Muestra la versión del kernel.
release
-n, --nodename    Muestra el nombre del equipo.
```

Ejemplo :

```
# uname
# uname -m
# uname -a
# uname -s
# uname -r
```

Comando dmidecode

Es una herramienta que permite conocer a fondo el hardware de nuestro equipo, tal como se describe en la BIOS del sistema según el SMBIOS / DMI estándar SMBIOS; el cual significa "System Management BIOS" y DMI significa "Desktop Management Interface" , pero el programa no llega esta allí nada mas. Esta herramienta es capaz de mostrar :

Los datos especificados en el SMBIOS se clasifican en tipos (types) y a cada uno de ellos se le identifica con un número en específico:

```
0 BIOS
1 System
2 Base Board
3 Chassis
4 Processor
5 Memory Controller
6 Memory Module
7 Cache
8 Port Connector
9 System Slots
10 On Board Devices
11 OEM Strings
12 System Configuration Options
13 BIOS Language
14 Group Associations
15 System Event Log
```

```
16 Physical Memory Array
17 Memory Device
18 32-bit Memory Error
19 Memory Array Mapped Address
20 Memory Device Mapped Address
21 Built-in Pointing Device
22 Portable Battery
23 System Reset
24 Hardware Security
25 System Power Controls
26 Voltage Probe
27 Cooling Device
28 Temperature Probe
29 Electrical Current Probe
30 Out-of-band Remote Access
31 Boot Integrity Services
32 System Boot
33 64-bit Memory Error
34 Management Device
35 Management Device Component
36 Management Device Threshold Data
37 Memory Channel
38 IPMI Device
39 Power Supply
```

Por tanto la sintaxis sería : **dmidecode -t número_de_type**

Opciones :

-h, --help	<i>Imprime la ayuda.</i>
-t, --type TYPE	<i>Muestra solamente un tipo determinado (cuadro anterior).</i>
-u, --dump	<i>No decodifica las entradas.</i>

Ejemplo :

```
# dmidecode -t 5,15
# dmidecode -u
```

<http://julio-linux.esdebian.org/32607/herramienta-dmidecode>

Comando lshw (bajar paquete)

Comando **lshw** (muestra información del sistema, discos, memorias, pci, usb... sería como ls del hardware.

Opciones :

-version	<i>Version del programa.</i>
-html	<i>Salida en formato html.</i>
-xml	<i>Salida en formato xml.</i>
-short	<i>Información del hardware.</i>

-businfo*Información del bus.*

Ejemplo :

```
# lshw -html  
# lshw -xml  
# lshw -short
```

Ejercicio :

- 1) Mostrar información del cpu.
- 2) Mostrar información de la memoria.
- 3) Obtener un listado de los pci.
- 4) Obtener un listado de los usb.
- 5) Información de la arquitectura.
- 6) Información de la version del kernel.

Resultados

- 1) cat /proc/cpuinfo
- 2) cat /proc/mem
- 3) lspci
- 4) lsusb
- 5) arch o uname -m
- 6) uname -r