

Capítulo IV: La consola y el nuevo árbol de directorios en Linux:

ÍNDICE DE CONTENIDOS:

- 4.1 La consola, nuestra herramienta de trabajo
- 4.2 Los principales directorios en Linux
- 4.3 Tipos de archivos en Linux:
 - 4.3.1 Intérprete de comandos I

Esta obra está protegida por la **Licencia Creative Commons**, bajo las condiciones de: **Reconocimiento - No comercial - Compartir igual**: El material creado por un artista puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

Reconocimiento (Attribution): El material creado por un artista puede ser distribuido, copiado y exhibido por terceras personas si se muestra en los créditos.

No Comercial (Non commercial): El material original y los trabajos derivados pueden ser distribuidos, copiados y exhibidos mientras su uso no sea comercial.

Compartir Igual (Share alike): El material creado por un artista puede ser modificado y distribuido pero bajo la misma licencia que el material original.

Versión: 1.0

Autor: Alberto Reynolds Moreno.
alberto.reynolds@gmail.com

Revisión: Isabel Rueda Rodríguez
rueda.isabel@gmail.com

Antes de comenzar a profundizar en el árbol de directorios de Linux, es buen momento para, a modo de prólogo, hablar muy superficialmente de una de las herramientas que como administradores de sistemas Linux vamos a utilizar a diario, se trata de **la consola o terminal**.

4.1 La consola, nuestra herramienta de trabajo:

Como ya hemos repetido varias veces, un buen administrador de Linux trabajará, la mayor parte de su tiempo, con una terminal de texto o consola desde donde ejecutará las órdenes y comandos necesarios para una buena administración del equipo.

La consola o terminal, será esa “*pantalla negra*” que nos aparecerá con un intérprete de comandos cuando no tenemos entorno gráfico o aquella ventana que, aún teniendo entorno gráfico, abriremos para ejecutar comandos.

La consola por defecto presenta una apariencia similar a la siguiente:

```
[albertux@debian]$
```

La información que nos ofrece esta terminal y su intérprete de comandos es la siguiente:

- **Nombre del usuario** con el que se ha logeado: en este caso “albertux”.
- **Nombre del host** que le hemos dado durante la instalación al equipo en el que estamos trabajando: en este caso “debian”.
- **El símbolo \$** que indica que estamos trabajando como un usuario final sin privilegios de administración.

***Nota:** cuando estemos trabajando como administrador, el símbolo \$ pasará a ser una #.*

De este modo sabremos si somos administrador o no desde una terminal.

Tras el símbolo \$ o #, tenemos preparado y paradeando nuestro intérprete de comandos, a través del cual, comenzaremos a pasarle los comandos y órdenes que necesitemos.

A lo largo de los capítulos iremos viendo algunos de los comandos fundamentales para poder administrar nuestra máquina Linux, lógicamente no servirá cualquier entrada en el intérprete:

```
[albertux@debian]$ hola
bash: hola: command not found
```

En el ejemplo anterior, la consola nos muestra la salida provocada al ejecutar el comando 'hola' sobre nuestro intérprete de comandos **bash**: 'command not found' (veremos más adelante que existen diferentes intérpretes de comandos).

Con esto tenemos una primera idea de lo que puede ser una consola o terminal y para qué vamos a utilizarla.

***Nota:** durante los sucesivos capítulos de este howto, indicaremos con el color verde la respuesta del intérprete de comandos al ejecutar una orden y con el color blanco el intérprete de comandos y la orden o comando a ejecutar.*

4.2 Los principales directorios en Linux:

Si somos usuarios noveles de Linux, quizás esta jerarquía de árbol de directorios nos pueda resultar algo confusa o desordenada, pero nada más lejos de la realidad, nos daremos cuenta a lo largo de este howto que cada elemento está perfectamente ubicado en un directorio concreto.



Fuente:www.tiraecol.net

Como hemos comentado, la estructura de árbol de directorios que ofrece Linux, puede ser, a primera vista, algo confusa; pero presume de ser de las estructuras de directorios más organizadas que se conocen.

Los principales directorios en Linux son:

DIRECTORIO	DESCRIPCIÓN
/	Es el directorio RAÍZ, la base del resto de directorios y por donde debe comenzar cualquier ruta.
/boot	Contiene los archivos del kernel y archivos de configuración del ARRANQUE del sistema.
/bin	Son los archivos EJECUTABLES ESENCIALES por todos los usuarios.
/dev	Directorio donde se encuentran los DISPOSITIVOS.
/etc	Directorio que contiene los archivos de CONFIGURACIÓN.
/etc/rc.d	Archivos de configuración de INICIALIZACIÓN.
/home	Directorio de los USUARIOS.
/lib	LIBRERÍAS esenciales y módulos del kernel.
/mnt	Directorio donde "MONTAR" los distintos dispositivos y/o particiones.
/mnt/cdrom	Directorio donde se monta el CD-ROM.
/mnt/floppy	Directorio donde monta la DISQUETERA.
/mnt/usb	Directorio donde monta un USB.
/proc	Información sobre partes del sistema (PROCESOS)
/root	Directorio del ROOT.

/sbin	Directorio que contiene los EJECUTABLES DEL ROOT.
/tmp	Directorio TEMPORAL
/usr	PROGRAMAS, documentación, fuentes... de los usuarios del sistema.
/var	Archivo de VARIABLES, logs...

Como se puede apreciar, la raíz del árbol es llamada con “/”, en Linux no existe c:, d:, a: ...

Otro tema a tener en cuenta, es que si disponemos de distintos discos duros podremos “montar” los distintos directorios en el disco que queramos, es decir, podremos instalar la mayoría de estos directorios en un primer disco duro maestro y, por ejemplo, los datos de usuario (*directorio HOME*) y el directorio de VAR en un segundo disco duro, como medida de seguridad para no perder estos datos en caso de pérdida del Sistema.

Este reparto será siempre transparente para la jerarquía de directorios. No deberemos indicar en que disco duro está el archivo o directorio al que quiero llegar como sucede en Windows (*con C: o D:*), sino que bastará con indicar simplemente la ruta; ej.: **cd /home/usuario**

Como usuario final, únicamente tendremos acceso y permisos, dentro de nuestro directorio personal ubicado en /home.

Si somos administrador (*root*), seremos responsables de la política de seguridad y mantenimiento del resto de directorios.

Recordatorio: como administrador (*root*) tendremos acceso y permisos para configurar y manipular cualquier directorio y archivo del sistema.

Un usuario que se valide como usuario final, sólo podrá “jugar” dentro de su directorio /home.

Lo realmente atractivo de Linux es ser ese administrador (*root*). Este Howto trata de la figura del administrador, así que estamos de suerte ;)

4.3 Tipos de archivos en Linux:

En Linux, existen cuatro tipos diferentes de archivos: archivos normales, especiales, directorios y enlaces.

- **Archivos normales:** pueden contener texto, código fuente en C, archivos de orden shell (programas, interpretados), programas binarios ejecutables...
- **Archivos especiales:** son archivos con una función especial en el sistema. Suelen hacer referencia a los dispositivos.
- **Los directorios:** incluyen en su interior otros archivos y subdirectorios. Los ficheros son identificados por el sistema por su número de **inodo**, el cuál es el único identificador del fichero para el sistema de ficheros.

Un directorio es una lista de números de inodo con sus correspondientes nombres de fichero. Cada nombre de fichero en un directorio es un enlace a un inodo particular.

- **Los enlaces:** existen dos tipos de enlaces, **enlaces duros** y **enlaces simbólicos**.

4.3.1 Intérprete de comandos I:

Visto lo anterior, comencemos a utilizar nuestro intérprete de comandos para realizar estas tareas.

Por ejemplo, supongamos que tenemos un fichero `'hola.txt'` en un directorio. Usando el comando `'ls -i'`, desde una terminal, veremos el número de inodo para ese fichero.

```
[albertux@debian]$ ls -i hola.txt
22192  hola.txt
```

En este ejemplo, el fichero `'hola.txt'` tiene el número de inodo `'22192'` en el sistema de ficheros.

Los enlaces duros: son un tipo de enlace que nos permite crear una especie de *'acceso directo'* entre un archivo o directorio original y otro, llamado enlace.

Mediante la orden `'ln'` ejecutada en nuestro intérprete de comandos desde una consola, podremos crear múltiples enlaces para un fichero.

Supongamos que queremos crear un archivo, al que llamaremos `'hola2.txt'` que será un enlace (*'acceso directo'*) a `'hola.txt'`:

```
[albertux@debian]$ ln hola.txt hola2.txt
```

Con `'ls -i'` veremos que los dos ficheros tienen el mismo inodo:

```
[albertux@debian]$ ls -i hola.txt hola2.txt
22192  hola.txt
22192  hola2.txt
```

Ahora, accediendo a uno o a otro accederemos al mismo fichero, ya que es el mismo inodo. Si haces cambios en `'hola.txt'`, estos cambios también serán efectuados en `'hola2.txt'` puesto que, a todos los efectos, son el mismo fichero.

Nota: sólo podemos crear enlaces duros entre ficheros del mismo sistema de ficheros, no sucede lo mismo con los enlaces simbólicos.

Un fichero sólo está definitivamente borrado del sistema cuando no quedan referencias a él. Puedo borrar `'hola.txt'` y seguirá existiendo `'hola2.txt'`.

Los enlaces simbólicos: los enlaces simbólicos son otro tipo de enlace, que es diferente al enlace duro. Un enlace simbólico permite dar a un fichero el nombre de otro, pero no enlaza el fichero con un inodo.

La orden `'ln -s'` crea un enlace simbólico a un fichero:

```
[albertux@debian]$ ln -s adios.txt adios2.txt
```

Crearemos un enlace simbólico `'adios2.txt'` apuntando al fichero `'adios.txt'`. Si usamos `'ls -i'`, veremos que los dos ficheros tienen inodos diferentes, en efecto:

```
[albertux@debian]$ ls -i adios.txt adios2.txt
22195  adios.txt
22341  adios2.txt
```

Si ahora borramos el original, *'adios.txt'*, el contenido de *'adios2.txt'* se perderá, aunque seguirá existiendo el archivo *'adios2.txt'* (vacío).