

ESTRUCTURA DEL SISTEMA DE ARCHIVOS DE LINUX

1. Tipos de archivos

La base del sistema de archivos de Linux, es obviamente el archivo, que no es otra cosa que la estructura empleada por el sistema operativo para almacenar información en un dispositivo físico como un disco duro, un disquete, un CDROM o un DVD. Como es natural un archivo puede contener cualquier tipo de información, desde una imagen en formato PNG o JPEG a un texto o una página WEB en formato HTML, ... El sistema de archivos es la estructura que permite que Linux maneje los archivos que contiene.

Todos los archivos de Linux tienen un nombre, el cual debe cumplir unas ciertas reglas:

- Un nombre de archivo puede tener entre 1 y 255 caracteres.
- Se puede utilizar cualquier carácter excepto la barra inclinada / y no es recomendable emplear los caracteres con significado especial en Linux, que son los siguientes: = \ ^ ~ ' " ` * ; - ? [] () ! & ~ < >.

Para emplear archivos con estos caracteres o espacios hay que introducir el nombre del archivo entre comillas.

Se pueden utilizar números exclusivamente si así se desea. Las letras mayúsculas y minúsculas se consideran diferentes, y por lo tanto no es lo mismo **carta.txt** que **Carta.txt** ó **carta.Txt**

Como en Windows, se puede emplear un cierto criterio de "tipo" para marcar las distintas clases de archivos empleando una serie de caracteres al final del nombre que indiquen el tipo de archivo del que se trata. Así, los archivos de texto, HTML, las imágenes PNG o JPEG tienen extensiones **.txt**, **.htm** (o **.html**), **.png** y **.jpg** (o **.jpeg**) respectivamente.

Pese a esto Linux sólo distingue tres tipos de archivos:

- Archivos o archivos ordinarios, son los mencionados anteriormente.
- Directorios (o carpetas), es un archivo especial que agrupa otros archivos de una forma estructurada.
- Archivos especiales, son la base sobre la que se asienta Linux, puesto que representan los dispositivos conectados a un ordenador, como puede ser una

impresora. De esta forma introducir información en ese archivo equivale a enviar información a la impresora. Para el usuario estos dispositivos tienen el mismo aspecto y uso que los archivos ordinarios.

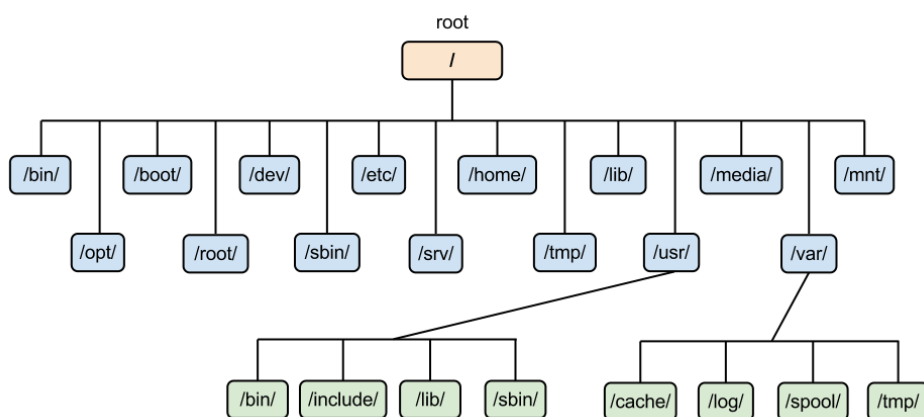
2. Herencias del Unix

El sistema de archivos de Linux sigue todas las convenciones de Unix, lo cual significa que tiene una estructura determinada, compatible y homogénea con el resto de los sistemas Unix.

Al contrario que en Windows o MS-DOS el sistema de archivos en cualquier sistema Unix no está ligado de una forma directa con la estructura del hardware, esto es, no depende de si un determinado ordenador tiene 1, 2 o 7 discos duros para crear las unidades c:\, d:\ o m:\.

Todos el sistema de archivos de Unix tiene un origen único la raíz o **root** representada por **/**. Bajo este directorio se encuentran todos los archivos a los que puede acceder el sistema operativo.

Estos archivos se organizan en distintos directorios cuya misión y nombre son estándar para todos los sistema Unix.



- /** Raíz del sistema de archivos.
- /dev** Contiene archivos del sistema representando los dispositivos que estén físicamente instalados en el ordenador.
- /etc** Este directorio está reservado para los archivos de configuración del sistema. En este directorio no debe aparecer ningún archivo binario (programas). Bajo este deben aparecer otros dos subdirectorios:
 - /etc/x11** Archivos de configuración de XWindow

<code>/etc/skel</code>	Archivos de configuración básica que son copiados al directorio del usuario cuando se crea uno nuevo.
<code>/lib</code>	Contiene las librerías necesarias para que se ejecuten los programas que residen en <code>/bin</code> (no las librerías de los programas de los usuarios).
<code>/proc</code>	Contiene archivos especiales que o bien reciben o envían información al kernel del sistema (Se recomienda no modificar el contenido de este directorio y sus archivos).
<code>/sbin</code>	Contiene programas que son únicamente accesibles al superusuario o <code>root</code> .
<code>/usr</code>	Este es uno de los directorios más importantes del sistema puesto que contiene los programas de uso común para todos los usuarios. Su estructura suele ser similar a la siguiente:
<code>/usr/X11R6</code>	Contiene los programas para ejecutar XWindow.
<code>/usr/bin</code>	Programas de uso general, lo que incluye el compilador de C/C++.
<code>/usr/doc</code>	Documentación general del sistema.
<code>/usr/etc</code>	Archivos de configuración generales.
<code>/usr/include</code>	Archivos de cabecera de C/C++ (<code>.h</code>).
<code>/usr/info</code>	Archivos de información de GNU.
<code>/usr/lib</code>	Librerías generales de los programas.
<code>/usr/man</code>	Manuales accesibles con el comando <code>man</code> .
<code>/usr/sbin</code>	Programas de administración del sistema.
<code>/usr/src</code>	Código fuente de programas.
	Existen además de los anteriores otros directorios que se suelen localizar en el directorio <code>/usr</code> , como por ejemplo los directorios de los programas que se instalen en el sistema.
<code>/var</code>	Este directorio contiene información temporal de los programas (lo cual no implica que se pueda borrar su contenido, de hecho, ¡no se debe hacer!)

3. Enlaces

Los enlaces son un tipo de archivo ordinario cuyo objetivo es crear un nuevo nombre para un archivo determinado. Una vez creado el enlace simbólico éste permite

acceder al archivo que enlaza de igual modo que si se hubiera copiado el contenido del mismo a otro archivo, con la ventaja de que este realmente no se ha copiado. Los enlaces simbólicos son especialmente útiles cuando se quiere que un grupo de personas trabajen sobre un mismo archivo, puesto que permiten compartir el archivo pero centralizan las modificaciones.

Como ejemplo se puede suponer la existencia de un archivo llamado **balance.1999.txt**, al que se crea un enlace simbólico **balance.txt**. Cualquier acceso a **balance.txt** es traducido por el sistema de forma que se accede al contenido de **balance.1999.txt**.

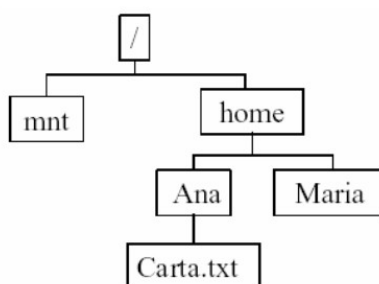
4. El camino o Path

En cualquier sistema operativo moderno la estructura de archivos es jerárquica y depende de los directorios. En general la estructura del sistema de archivos se asemeja a una estructura de árbol, estando compuesto cada nudo por un directorio o carpeta, que contiene otros directorios o archivos.

En Windows cada unidad de disco se identifica como una carpeta básica que sirve de raíz a otras, y cuyo nombre es especial a:, c:, d: etc. En los sistemas Unix, y por lo tanto en Linux, existe una única raíz llamada / de la que cuelgan todos los archivos y directorios, y que es independiente de qué dispositivos estén conectados al ordenador.

El camino o **path** de un archivo o directorio es la secuencia de directorios que se ha de recorrer para acceder a un determinado archivo separados por /.

Observemos el siguiente ejemplo:



Existen dos formas del **path** o camino:

1. El camino absoluto que muestra toda la ruta a un archivo, **/home/Ana/Carta.txt**.

2. El **path** relativo a un determinado directorio, por ejemplo si no encontramos en el directorio **/home**, el **path** relativo al archivo **Carta.txt** es **Ana/Carta.txt**

Para complicar aun más las cosas, todos los directorios contienen dos directorios especiales:

- El directorio actual, representado por el punto **.**
- El directorio padre representado por dos puntos **..**

Estando en el directorio **/home/Maria** se puede acceder a **Carta.txt** con **/home/Ana/Carta.txt** (**path** absoluto) o bien **../Ana/Carta.txt** (**path** relativo). En Ana como **./Carta.txt** o simplemente **Carta.txt**.

Acceso a los diferentes sistemas de archivos

Como se ha visto anteriormente el sistema de archivos de Linux sólo tiene una raíz y su estructura es independiente de los dispositivos de almacenamiento existentes. Esto implica que el procedimiento a emplear para acceder a la información almacenada en los distintos sistemas de almacenamiento de un ordenador no es tan sencilla como en Windows, y requiere un proceso llamado "**montado**", con la orden **mount**. Cuando se ha terminado de trabajar con un determinado dispositivo hay que "**desmontarlo**".

Veamos un ejemplo desde la línea de comandos de cómo accederíamos a una disquetera:

1. Introducimos el disquete en la disquetera.
2. Montamos el sistema de archivos que en el caso de una disquetera sería mediante la siguiente orden:

```
mount -t vfat /dev/fd0 /mnt/floppy
```

3. Leemos, grabamos, y manipulamos el contenido del disquete.
4. Desmontamos el sistema de archivos del disquete con la orden **umount**.
5. Extraemos el disquete de la disquetera.

El proceso puede parecer complejo pero es el precio a pagar por la seguridad, puesto que de esta forma se garantiza que no exista ninguna aplicación que esté usando el disquete cuando se extraiga.

La orden para montar un CD-ROM en Linux desde la línea de comandos es la siguiente:

```
mount -t iso9660 /dev/hdb /mnt/cdrom
```

En el caso de los CD-ROM Linux impide su extracción hasta que se desmonta.

Para complicar más las cosas sólo el administrador o **root** tiene permiso para montar y desmontar un sistema de archivos (por motivos de seguridad), aunque esto puede ser arreglado. Se debe de tener en cuenta que esto con los entornos gráficos está mucho más automatizado y es más fácil su manejo como ya veremos.

5. Permisos

Linux, al igual que todos los sistemas derivados del Unix, mantiene un sistema de permisos de acceso a los archivos muy estricto, a fin de controlar qué es lo que se puede hacer con ellos, y quien lo puede hacer. Estos permisos se identifican con letras y son:

- r** permiso de **lectura** el archivo
- w** permiso de **escritura** en el archivo
- x** permiso de **ejecución** del archivo
- s** permiso para **cambiar el propietario** del archivo

Al contrario que en Windows o MS-DOS los programas ejecutables de Linux no están marcados por una determinada extensión (**.exe**) sino por un atributo, el permiso de ejecución **x**. Si se elimina este atributo a un programa, Linux no será capaz de ejecutarlo.

A su vez cada uno de estos permisos se aplica: al dueño del archivo (**u**), al grupo de usuarios al que pertenece el dueño (**g**), ó al resto de usuarios (**a**). Así un archivo determinado puede tener permiso para ser leído, escrito y ejecutado por su dueño, leído y ejecutado por el grupo al que pertenece y no tener ningún tipo de acceso para los demás usuarios. Como se puede entender este tipo de mecanismo es especialmente útil cuando se trabaja en grupo en un determinado proyecto.

6. Compatibilidad de LINUX

La compatibilidad ha sido y será uno de los grandes éxitos de Linux, y en esto tiene mucho que ver la gente que está colaborando en el desarrollo de Linux, que trabaja habitualmente sobre distintas plataforma software, motivo este, que causa ese interés razonado por mantener la compatibilidad con cualquier otro entorno informático.

Si trabajáramos con Mac o con Pc's, nos interesaría, en un momento dado, el poder intercambiar la información que pudiera tener en ambos equipos. En Linux, estas necesidades van siendo solventadas sobre la marcha, de manera que hoy día, se

puede decir que alrededor del 50% del software producido a nivel mundial se podría ejecutar sobre una plataforma Linux.

Actualmente contamos con emuladores para los principales entornos 'domésticos' del mundo, el '**Dosemu**' para Ms-Dos, el '**wabi**' y el '**wine**' para Windows.

Aparte de las emulaciones, que como es lógico y normal, no rinden igual que el sistema operativo al cual emulan, contamos con el '**samba**', que nos permite interactuar con sistemas Win98/2000 utilizando su famoso '**Netbios**', logrando compartir recursos de forma transparente en cualquier pequeña, o no tan pequeña, red de área local.

También hemos logrado lo mismo con los entornos Mac, donde podemos compartir recursos de forma transparente. Asimismo, hemos logrado interconectar los servidores de red Novell con cualquier servidor Linux, de forma tal que podemos montar nuevos volúmenes desde Linux sin problemas.

Entre los formatos de archivos soportados por Linux tenemos los siguientes:

Sistemas de archivos	Tipo	Definición
Extenden File System	Ext	Reemplazado por ext2
Second Extenden FileSystem	Ext2	Archivos más común de Linux
Minix FileSystem	minix	Archivos Minix original
Xia Filesystem	Xia	Como Ext2 (de poco uso)
Umsdos Filesystem	umsdos	Para instalar Linux en particiones MS-DOS
MS-DOS File System	msdos	Utilizados para acceder a archivos MS-DOS
/proa File System	proc	Da información sobre procesos ps , etc.
ISO 9660 File System	Iso9660	Formato utilizado por muchos CD - ROMs
Xenix File System	xenix	Sistemas de archivos Xenix
System V File System	sysv	Variantes del System V para el x86
Coherent File System	coherent	Acceso de archivos de coherent
HPFS File System	hpfs	Acceso de lectura en particiones HPFS

7. Descompresión de archivos con extensión **.tgz** o **.tar.gz**

El nombre de la extensión se debe a que en su creación han intervenido dos programas **tar** y **gzip**, el primero para recopilar archivos y directorios en un archivo y el segundo para comprimir dicho archivo. Lo que se suele hacer en primer lugar, antes de descomprimirlo es ver los archivos que contiene y sobre todo que

estructura de directorios posee, esto se realiza (tomando el archivo `netscape.tgz`) con la orden

```
tar -ztfv netscape.tgz | more.
```

Si la estructura de directorios nos convence y queremos descomprimir el archivo en el directorio actual, la orden es:

```
tar -zxvf netscape.tgz
```

Por contra, si queremos hacerlo en un directorio distinto del actual lo haremos indicándolo con la opción `-C`, esto es:

```
tar -C /tmp -zxvf netscape.tgz
```

Actualmente gracias a entornos gráficos como KDE y Gnome la mayor parte de estas operaciones pueden realizarse con simples clics de ratón. Sin embargo nunca está demás aprenderse para lograr la independencia del sistema gráfico y para sacarnos de algún que otro apuro.

8. Ejercicios

1. Cree un archivo `.tar.gz` de un directorio pequeño (por ejemplo cree uno de prueba con al menos dos archivos).

Solución

```
tar cvfz ej.tar.gz ej/
```

2. ¿En su directorio personal hay archivos de configuración? En caso afirmativo cite algunos.
3. Estando en un intérprete de comandos vaya al directorio `/usr/share/pixmaps` y liste con `ls` y caracteres comodín todos los archivos con extensión `.xpm`. Desde `bash`, copie uno de esos archivos a su directorio con el comando `cp`. (Ayuda: busque ayuda sobre el comando `cp` y emplee `~` para referirse a su directorio).

Solución

```
cd /usr/share/pixmaps
```

```
ls *.xpm
```

```
cp arc.xpm ~
```

4. Para mejorar la seguridad de sus archivos y evitar que otras personas puedan consultarlos, quite a su directorio el permiso de ejecución para el grupo y para otros usuarios.

Solución

```
cd ~ ; chmod og-x
```

5. Revise los directorios de los usuarios del sistema en **/home**. Identifique y liste directorios con permisos de seguridad deficientes. Intente cambiar los permisos de archivos o directorios de otro usuario.

Solución

```
cd /home ; ls -l ; chmod -w miriam
```