

ARCHIVO

Toda la información, ya sean textos, imágenes, o información para la configuración del sistema, se almacena en "ficheros", que a su vez se guardan en "directorios". Los ficheros son la estructura empleada por el sistema operativo para almacenar información en un dispositivo físico como un disco duro, un disquete, un CD-ROM. Con todas las herramientas y programas existentes se puede acceder a estos ficheros para ver su contenido o modificarlo.

La base del sistema de archivos de Linux, es obviamente el archivo o fichero.

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 caracter excepto la barra inclinada (/). No es recomendable emplear los siguientes caracteres: = \ ^ ~ ' " ` * ; - ? [] () ! & ~ < >, ya que éstos tienen un significado especial en Linux.

Nota: Para emplear ficheros con estos caracteres o espacios hay que introducir el nombre del fichero 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. Se puede emplear un cierto criterio de "tipo" (extensión) para marcar las distintas clases de ficheros empleando una serie de caracteres al final del nombre que indiquen el tipo de fichero del que se trata. Así, los ficheros 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 ficheros ordinarios:** Son los mencionados anteriormente.
- **Directorios (o carpetas):** Es un archivo especial que agrupa otros ficheros 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.

DIRECTORIO

Un directorio es un conjunto de archivos, que a su vez pueden contener otros directorios. Los directorios también poseen permisos, y eso es muy importante para evitar que los usuarios sin experiencia, borren o modifiquen algo que no deban. El árbol de directorios nos ayuda a saber dónde se encuentra un archivo.

A diferencia de los sistemas **MS-DOS/Windows**, en los sistemas UNIX no se dividen los directorios entre las distintas unidades físicas (C:, D:, etc.). Al contrario, para UNIX todos son ficheros y directorios. Éste considerará un disco duro o una disquetera como directorios. Por tanto, no tiene sentido escribir en la consola "C:", sino que nos iremos al directorio asociado a esa partición.

Todos los ficheros y directorios de un sistema UNIX cuelgan de un directorio principal llamado "**raíz**", que se representa como `/`. El directorio raíz es la base para todo el árbol de directorios, es allí donde están contenidos todos los directorios del sistema.

Cuando el usuario accede a una sesión, Linux "envía" al usuario a su directorio de trabajo, que es su directorio personal (`/home/nombre-usuario`). En donde el usuario tiene la libertad absoluta para hacer lo que quiera con sus ficheros y directorios ubicados ahí. Sin embargo no podrá hacer todo lo que quiera en el directorio de otro usuario, ya que Linux tiene un sistema de permisos que concede o restringe libertades sobre los directorios y ficheros que hay en Linux. Aunque existe un usuario "**root**", que sí tiene permiso de hacer cualquier cosa en Linux.

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 nodo por un directorio o carpeta, que contiene otros directorios o archivos.

El camino o path de un fichero o directorio es la secuencia de directorios que se ha de recorrer para acceder a un determinado fichero separados por `/`. Supongamos la estructura de archivos de la siguiente figura:

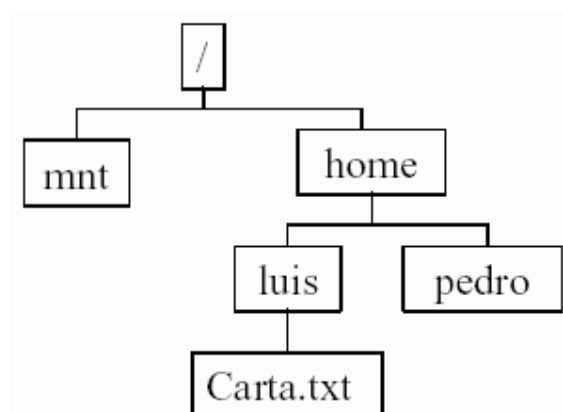


Figura 2

Existen dos formas del path o camino:

- el camino absoluto que muestra toda la ruta a un fichero, `/home/luis/Carta.txt`.

- el path relativo a un determinado directorio, por ejemplo si no encontramos en el directorio */home*, el path relativo al fichero *Carta.txt* es *luis/Carta.txt*

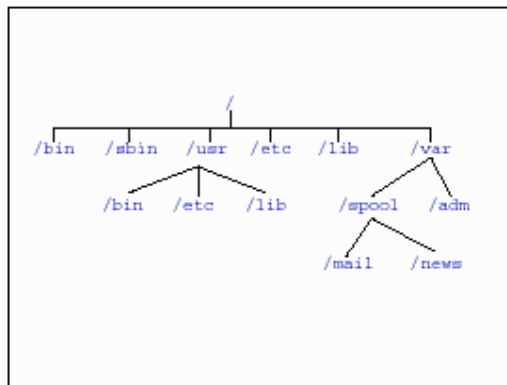
Para complicar aun más las cosas, todos los directorios contienen dos directorios especiales:

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

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

ESTRUCTURA DEL SISTEMA DE ARCHIVOS DE LINUX

El sistema de ficheros es un conjunto de archivos que se encuentran organizados en un árbol, como se observa en la siguiente figura.



El sistema de archivo 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. 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 ó 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 ficheros a los que puede acceder el sistema operativo. Estos ficheros se organizan en distintos directorios cuya misión y nombre son estándar para todos los sistema Unix.

El Sistema Linux tiene un sistema de directorios muy organizado, donde cada uno tiene una función, a continuación se definen dichas funciones:

/boot Este directorio contiene la información necesaria para poder arrancar nuestro sistema GNU/Linux. Entre otros archivos, aquí se encuentran los núcleos que se pueden iniciar (normalmente se trata de archivos con un nombre similar a *vmlinuz-x.y.z*).

/etc Este directorio contiene todos los archivos de configuración de nuestro sistema GNU/Linux (como el archivo *passwd*, que contiene una lista con los nombres de usuario y contraseñas, o el archivo *fstab*, que contiene una lista con los puntos donde se han montado las diferentes particiones y los sistemas de archivos que utilizan).

La jerarquía /etc

Este directorio posee distintos subdirectorios que se utilizan para la configuración de determinados componentes de nuestro sistema:

/etc/opt Contiene los archivos de configuración para el software instalado en el directorio /opt. Cada paquete de software tendrá su correspondiente subdirectorio dentro de */etc/opt*.

/etc/X11 Contiene los archivos de configuración del entorno gráfico *X Window*. Entre otras cosas, contiene los gestores de ventanas y de arranque de los distintos escritorios gráficos. Este directorio es opcional.

/etc/sgml Contiene los archivos de configuración para SGML y XML. Este directorio es opcional.

/etc/skel Ficheros de configuración básica que son copiados al directorio del usuario cuando se crea uno nuevo.

/etc/fstab Contiene información de las diferentes particiones de disco o dispositivos que se pueden montar en un momento dado.

/etc/passwd Contiene información sobre los usuarios, como login, nombre y otra información que el administrador quiera agregar.

/etc/shadow Almacena los password en forma encriptada y las fechas de expiración de los mismos.

/etc/services Tabla en la cual se describen los servicios a los que tienen acceso los usuarios y por qué puertos trabajan.

/etc/xinetd.d Archivo para configurar los demonios.

/etc/rc.d/init.d Contiene información para inicializar los demonios.

/etc/hosts.deny Almacena información para configurar los permisos sobre los servicios que ofrece.

/etc/hosts.allow También información para configurar los permisos sobre los servicios que ofrece.

/bin Contiene programas ejecutables (también llamados *binarios*) que forman parte del sistema operativo GNU/Linux. Estos comandos pueden ser usados por cualquier usuario y son relativos a la consola.

/sbin Contienen programas ejecutables (también llamados *binarios*) que forman parte del sistema operativo GNU/Linux. Estos comandos son relativos a los sistemas de archivos, particiones e inicio del sistema, y solo pueden ser usados por el administrador.

/lib Contiene las bibliotecas (o librerías) del sistema que son necesarias durante el inicio del mismo. Estas bibliotecas son análogas a los archivos DLL de Windows. Su ventaja reside en que no es necesario integrar su código en los programas que las usan, ya que cuando un programa necesita alguna de sus funciones, se carga la biblioteca en la memoria y puede ser usada por cualquier otro programa que la necesite, sin necesidad de volver a cargarla en memoria. Un subdirectorio especial es */lib/modules*, que contiene los módulos del núcleo (normalmente se trata de controladores de dispositivos) que se cargan únicamente en caso de que haga falta usar un determinado dispositivo, por lo que no estarán permanentemente ocupando memoria.

/mnt Este directorio es típico de las distribuciones *RedHat*, aunque puede no estar presente en otras distribuciones. Su misión consiste en agrupar en un mismo lugar los puntos de montaje de diversas particiones externas, como por ejemplo: CD-ROM, DVD, disqueteras, unidades ZIP, particiones de MS-Windows y de otros sistemas operativos, etc. Este directorio contiene un subdirectorio adicional para cada una de estas particiones (como */mnt/cdrom*, */mnt/win_c*, */mnt/floppy* o */mnt/zip*). Si accedemos a estos subdirectorios estaremos accediendo realmente a esas particiones.

/usr Su nombre proviene de *user* (*usuario*) y, como tal, contiene una réplica de otros directorios de nuestro sistema GNU/Linux orientados al usuario en lugar de al propio sistema. Este directorio se puede compartir en una red local, pero su contenido es estático.

/opt Contiene paquetes de software adicionales, como por ejemplo KDE o GNOME. Algunas distribuciones no hacen uso de él.

/var Su nombre procede de *variable*, y esa es la naturaleza de la información que contienen sus subdirectorios y archivos, como colas de impresión (en */var/spool/lpd*), correo electrónico que todavía no se ha recogido (en */var/spool/mail*) o enviado (en */var/spool/mqueue*), o archivos de registro creados por los distintos procesos de nuestro sistema (en */var/log*).

/tmp Este directorio contiene diversos archivos temporales que son usados por distintos programas de nuestro sistema.

/dev Este directorio contiene archivos de dispositivos que permiten la comunicación con los distintos elementos hardware que tengamos instalados en nuestro sistema.

Entre los distintos dispositivos que aquí podemos encontrar están los discos duros (como `/dev/hda` o `/dev/sda`), las particiones de los discos duros (como por ejemplo `/dev/hda1` o `/dev/sda4`), las unidades de CD-ROM SCSI (como `/dev/scd0`) y los vínculos a cualquier tipo de unidad de CD-ROM (como `/dev/cdrom`), las disqueteras (como `/dev/fd0`), las impresoras (como `/dev/lp0`), los puertos serie (como `/dev/ttyS0` o `/dev/cua0`), el puerto PS/2 (`/dev/psaux`) y las tarjetas de sonido (como `/dev/audio`).

/proc Contiene los *archivos* del sistema de archivos de proceso. No son verdaderos archivos, sino una forma de acceder a las propiedades de los distintos procesos que se están ejecutando en nuestro sistema. Para cada proceso en marcha existe un subdirectorio `/proc/<número de proceso>` con información sobre él.

/root Este es el directorio personal del usuario *root* o superusuario. Contiene básicamente la misma información que los directorios personales de los distintos usuarios del sistema, pero orientada única y exclusivamente al usuario *root*.

/home Aquí residen los directorios de los distintos usuarios del sistema, excepto el del usuario *root*, que se encuentra en el directorio especial */root*. Este directorio es opcional, aunque está presente en casi todas las distribuciones.

En un sistema GNU/Linux, cada usuario dispone de su propio directorio personal donde puede guardar los documentos creados por él con los distintos programas. Este directorio personal puede ser de acceso exclusivo para cada usuario, por lo que ningún otro usuario podrá entrar en él y visualizar el contenido de los archivos que contiene (aunque esto depende del nivel de seguridad seleccionado durante la instalación del sistema, ya que lo normal suele ser que el resto de usuarios puedan entrar en él y ver el contenido de algunos archivos y subdirectorios, pero no modificarlo).

Los directorios personales están ubicados en `/home/<nombre de usuario>`. El usuario *root* sí dispone de los permisos suficientes para acceder a los directorios personales del resto de usuarios.

La jerarquía **/usr**

Este directorio contiene distintos subdirectorios:

/usr/bin Contiene casi la totalidad de los comandos que un usuario puede necesitar.

/usr/include Contiene los archivos de cabeceras usados por los programas escritos en el lenguaje C.

/usr/lib Contiene las bibliotecas de usuario.

/usr/local Contiene la jerarquía local de usuario, con los subdirectorios típicos */bin*, */include*, */lib*, */sbin*, etc. Tras la instalación de un sistema GNU/Linux, este directorio estará vacío.

/usr/sbin Contiene los archivos binarios (ejecutables) no vitales del sistema usados por el administrador.

/usr/share Contiene archivos de datos independientes de la arquitectura del sistema, como documentación, *howtos*, etc. Aquí se encuentran los subdirectorios */man* (que

contiene manuales de programas) y */misc* (que contiene diversos datos independientes de la arquitectura). También se pueden encontrar los subdirectorios */dict*, */doc*, */games*, */info*, */locale*, */smgl*, *zoneinfo*, etc.

/usr/X11R6 Contiene todos los programas que componen el entorno gráfico *X Window* (binarios, bibliotecas y páginas *man*, por lo que encontraremos directorios como */usr/X11R6/bin* */usr/X11R6/lib* y */usr/X11R6/man*). Este directorio es opcional.

/usr/games Contiene los archivos binarios de muchos juegos y programas educativos. Este directorio es opcional.

/usr/src Contiene el código fuente del núcleo de GNU/Linux. Este directorio es opcional.

/usr/local Como en el resto de sistemas operativos UNIX, GNU/Linux está diseñado para ser usado en entornos de red, por lo que es bastante frecuente que el directorio */usr* no se encuentre en el ordenador local, sino en un servidor. Por ello, el directorio */usr/local* está destinado a contener el software que se instale localmente en cada ordenador, y contiene subdirectorios similares a los que se puede encontrar bajo */usr* (como */usr/local/bin*, */usr/local/sbin*, */usr/local/lib*, */usr/local/man*, etc.). Adicionalmente, encontraremos un directorio */usr/local/etc* con idéntica funcionalidad al directorio */etc*. Aunque use su ordenador de forma aislada y no en un entorno de red, este es el directorio en el que debería instalar sus programas.

La jerarquía */var*

Este directorio también posee una estructura típica de subdirectorios:

/var/cache Contiene datos de aplicaciones en *cache*, como archivos del servidor web y páginas de manual.

/var/lib Contiene información de estado variable.

/var/log Contiene registros creados por diversos programas y por el propio sistema operativo.

/var/run Contiene datos relevantes para algunos procesos que se están ejecutando.

/var/spool Contiene datos encolados de diversas aplicaciones, como correo electrónico del sistema y colas de impresión.

/var/tmp Contiene datos temporales que deben ser preservados entre reinicios del sistema.

/var/spool/mail Archivos de correo de los distintos usuarios.

Otros directorios

Algunas distribuciones hacen uso de otros directorios adicionales que suelen montar en el directorio raíz:

/media Es un directorio análogo al estándar */mnt*, y contiene los puntos de montaje de los dispositivos extraíbles.

/windows Algunas distribuciones lo crean si detectan un sistema Microsoft Windows instalado en nuestro equipo, creando dentro de éste tantos subdirectorios como particiones de Windows tengamos (como */windows/C*, */windows/D*, etc.). Contiene el punto de montaje de estas particiones.

/srv Aquí se alojan los directorios que usan los distintos servidores que tengamos instalados, como el servidor web (en */srv/www*) o el servidor FTP (en */srv/ftp*).

Estos directorios no forman parte del estándar FHS, por lo que deberían ser evitados.

PERMISOS SOBRE ARCHIVOS Y DIRECTORIOS

En cualquier sistema multiusuario, es preciso que existan métodos que impidan a un usuario no autorizado copiar, borrar, modificar algún archivo sobre el cual no tiene permiso.

En Linux las medidas de protección se basan en que cada archivo tiene un propietario (usualmente, el que creó el archivo). Además, los usuarios pertenecen a uno o mas grupos, los cuales son asignados por el Administrador dependiendo de la tarea que realiza cada usuario; cuando un usuario crea un archivo, el mismo le pertenece también a alguno de los grupos del usuario que lo creó.

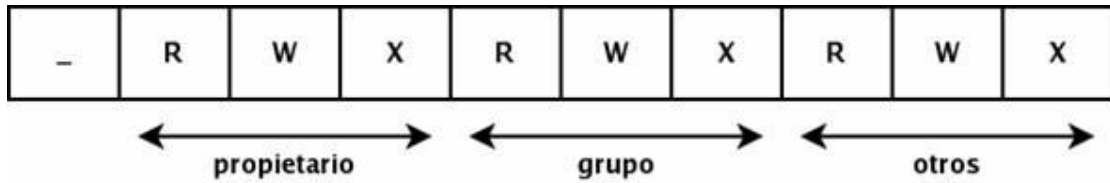
Así, un archivo en Linux le pertenece a un usuario y a un grupo, cada uno de los cuales tendrá ciertos privilegios de acceso al archivo. Adicionalmente, es posible especificar que derechos tendrán los otros usuarios, es decir, aquellos que no son el propietario del archivo ni pertenecen al grupo dueño del archivo.

En cada categoría de permisos (usuario, grupo y otros) se distinguen tres tipos de accesos: lectura (**R**ead), escritura (**W**rite) y ejecución (**eX**ecute), cuyos significados varían según se apliquen a un archivo o a un directorio.

En el caso de los archivos, el permiso R (lectura) habilita a quién lo posea a ver el contenido del archivo, mientras que el permiso W (escritura) le permite cambiar su contenido. El permiso X (ejecución) se aplica a los programas y habilita su ejecución.

Para los directorios, el permiso R permite listar el contenido del mismo (es decir, "leer" el directorio, mientras que el W permite borrar o crear nuevos archivos en su interior (es decir, modificar o "escribir" el directorio). El permiso X da permiso de paso, es decir, la posibilidad de transformar el directorio en cuestión en el directorio actual (ver comando *cd*).

En los listados de directorio, los permisos se muestran como una cadena de 9 caracteres, en donde los primeros tres corresponden a los permisos del usuario, los siguientes tres a los del grupo y los últimos, a los de los demás usuarios. La presencia de una letra (*r*, *w* o *x*) indica que el permiso está concedido, mientras que un guión (-) indica que ese permiso está denegado.



Permisos de ficheros

Usaremos el término fichero en su sentido más amplio. Es decir que el tema de permisos es aplicable a distintos tipos de ficheros con algunas matizaciones que explicaremos más adelante. Los ficheros tienen muchos atributos además de su nombre. Para ver los más significativos haremos:

```
$ ls -l /
```

Supongamos que tenemos el siguiente fichero llamado 'kkkkk'

[illegible]

En inglés se usan los términos owner , group, y others para designar respectivamente al propietario, al grupo y a cualquier usuario.

Notación numérica para permisos

Recuerde que también tenemos que pensar en la posibilidad de que mi amigo David acostumbrado al uso del azadón decida hacer este curso y teníamos un compromiso de explicar las cosas partiendo de cero. Los permisos de los ficheros son almacenados en formato binario y se puede referenciar numéricamente. Vimos que a cada permiso individual le asociábamos un número de tres dígitos formado por dos ceros y un tercer número que podía ser únicamente 1, 2, o 4. Por ejemplo el permiso de escritura para un usuario cualquiera era 002. Con estos números se puede codificar los permisos de la forma que indicamos en el siguiente ejemplo:

r w x - - w x - r - x Esto equivaldría a un permiso 735
4 2 1 - 0 2 1 - 4 0 1 (4+2+1 , 0+2+1 , 4+0+1 = 7,3,5)

Los permisos también pueden representarse como una secuencia de bits. Un bit es un valor que solo puede valer 0 o 1. En el caso anterior podríamos representarlo de la

forma 111 010 101 donde 1 indica que si hay permiso 0 que no lo hay y la posición de cada 0 y cada 1 representa a cada uno de los permisos.

Enlace Simbólico

```
lrwxrwxrwx  1 root    root      8 Oct 13 13:45 juan -> jdbotero
```

INODO

Un INODO es una estructura de datos, por así decirlo una tabla que contiene información sobre un fichero.

Cada fichero se identifica por un número de inodo. Este número es único dentro de todo el sistema de ficheros.

Dentro de cada inodo existe la siguiente información:

- Número de inodo
- Tipo de fichero
- Propietario de dicho fichero
- Permisos del fichero
- Fecha de creación del mismo

Puedes ver toda esta información si ejecutas un `ls -l` en cualquier directorio de tu sistema.

La tabla con todos los inodos se crea en el arranque del sistema.

El comando `find` utiliza esta tabla para hacer sus búsquedas.

Diferencia enlace simbólico y rígido

Enlaces de archivos

En ocasiones es necesario dar a un mismo archivo distintos nombres o, para mantener la integridad de los datos, se hace necesario que un mismo archivo que va a ser modificado por varias personas pueda estar representado por un nombre distinto con distintos permisos cada uno. Dada la representación que GNU/Linux le da a los archivos es posible que dos o más nombres apunten al mismo contenido en el disco rígido. Recordemos que los archivos se representan por el número de inodo en el disco y es el único identificador que el sistema tiene del archivo. Si se quiere ver el inodo que representa un archivo basta solo con agregar el modificador `-i` al comando `ls`. Existen dos tipos de enlaces: los duros y los simbólicos.

Enlaces duros o hard links

Para crear enlaces entre los archivos se utiliza el comando **ln**. Para dar un ejemplo, si se quisiera hacer un enlace del archivo hola el comando es

```
[shrek@pantano:~]$ ln hola saludo
```

Si ahora hacemos un listado mostrando los inodos

```
[shrek@pantano:~]$ ls -il
total 2
14438 -r----- 2 shrek  user    64 Apr 29 14:04 hola
14438 -r----- 2 shrek  user    64 Apr 29 14:04 saludo
```

Como se ve el número de nombres que los archivos tienen ahora es 2 dado que hemos creado otro nombre que es un enlace al contenido que tenía el archivo hola en el disco. Esto puede verse dado que el primer número del listado, 14438, es el número del inodo donde comienza el contenido del archivo hola. Aunque parezca que el contenido se ha duplicado, esto no es así. Ambos archivos apuntan al mismo sitio. Por esto, un archivo no se borra hasta que se haya borrado el último de sus nombres. De esta forma también podemos decir que un directorio no es más que un archivo que contiene información sobre la translación enlace a inodo. También que cada directorio tiene dos enlaces duros en él: `.` (un enlace apuntando a si mismo) y `..` (un enlace apuntando a su directorio padre). En el directorio raíz `/` el enlace `..` siempre apunta a `/`.

Enlaces simbólicos o symbolic links

A diferencia con los enlaces duros, estos enlaces solo dan otro nombre a un archivo pero no hacen un enlace al nivel de inodo. Se puede hacer una comparación con los "Accesos directos" de Windows95. La orden `ln -s` genera un enlace simbólico.

```
[shrek@pantano:~]$ ln -s hola saludos
[shrek@pantano:~]$ ls -il
total 2
14438 -r----- 1 shrek  user   64 Apr 29 14:04 hola
14446 lrwxrwxrwx 1 shrek  user    4 May  7 08:33 saludos -> hola
[shrek@pantano:~]$
```

Como se ve, el enlace nos muestra a que archivo está apuntando. Existen diferencias entre este tipo de enlaces y los duros como que ya no están apuntando al mismo inodo. Además los bits de permisos aparecen todos encendidos dado que no son usados. Los permisos que se utilizan son los del archivo al que se está apuntando. Si nos fijamos bien, en la primera posición de los permisos veremos una letra `l`. Esto nos indica que es un enlace simbólico. Otra particularidad es que se pueden crear enlaces simbólicos de archivos que no existen; pero lo mismo no es cierto para los enlaces duros. Con los enlaces simbólicos se puede saber a que archivo están apuntando, no así con los duros. Los enlaces simbólicos son ampliamente usados para las librerías compartidas.

Ficheros especiales

En unix todo es tratado como un fichero, y tanto los dispositivos como los directorios no son una excepción. Veamos algunos de los más importantes:

nombre	descripción
.	nombre del directorio en que nos encontramos
..	nombre del directorio anterior
~	nombre del directorio raíz del usuario
/dev/null	el dispositivo nulo
/dev/dsp	el dispositivo de sonido
/dev/*	dispositivos del sistema
/proc/*	información sobre el sistema