SISTEMAS INFORMÁTICOS UD-3

SISTEMAS OPERATIVOS. GESTION DE ARCHIVOS Y ALMACENAMIENTO

ESTRUCTURA DE DIRECTORIOS LINUX

ÍNDICE

[INTRODUCCIÓN 3](#_Toc125213234)

[ESTÁNDAR FHS (FILESYSTEM HIERARCHY STANDARD) 3](#_Toc125213235)

[DISTRIBUCIÓN DE DISCOS Y PARTICIONES EN EL SISTEMA DE FICHEROS FHS 4](#_Toc125213236)

[VENTAJAS DE MONTAJE EN VARIAS PARTICIONES 4](#_Toc125213237)

[INODOS 5](#_Toc125213238)

[ENLACES DUROS Y ENLACES BLANDOS 5](#_Toc125213239)

# INTRODUCCIÓN

La estructura de directorios de un SO determina cómo puede utilizar el usuario los archivos.

Se parte de un directorio origen (raíz) que a partir de ahí surgen varios directorios hijos (que pueden ser padres de nuevos directorios origen) dando lugar a un sistema de árbol.

Por ejemplo, el Escritorio en Windows es un directorio que parte desde C: siguiendo este esquema:

C: 🡪 Usuarios 🡪 NombreUsuario 🡪 Escritorio

En el caso de Linux:

/home/nombreUsuario/Escritorio

Del Escritorio a su vez pueden pender nuevos directorios hijos.

# ESTÁNDAR FHS (FILESYSTEM HIERARCHY STANDARD)

En Linux utilizamos un sistema de ficheros ext (actualmente ext4) el cual utiliza un estándar llamado FHS que se compone de los siguientes directorios:

* **/**: Directorio raíz.
* **/boot**: Ficheros de arranque.
* **/bin**: Ficheros ejecutables.
* **/sbin**: Ficheros ejecutables sólo por el administrador.
* **/lib**: Bibliotecas de programas compartidos (para que los programas puedan realizar sus acciones, por ejemplo, crear ventanas).
* **/usr**: Programas del ordenador.
* **/user/local**: Ficheros que instala localmente el administrador.
* **/opt**: Es igual que el anterior, pero para paquetes que no vienen con el SO.
* **/home**: Datos de los usuarios.
* **/etc**: Archivos de configuración de sistema.
* **/root**: Directorio del usuario root (administrador).
* **/var**: Ficheros efímeros. Ficheros donde se almacenan errores, registros de uso… los utiliza el propio SO.
* **/tmp**: Archivos temporales de los programas.
* **/mnt**: Alberga el montaje de dispositivos.
* **/media**: Montaje de discos extraíbles.
* **/dev**: Ficheros que hacen de interfaces de hardware (por ejemplo, el HDD).
* **/proc**: Sistema de ficheros virtual.

DISTRIBUCIÓN DE DISCOS Y PARTICIONES EN EL SISTEMA DE FICHEROS FHS

Los discos duros en Linux se sitúan en el directorio /dev. Los discos se llaman por una letra (a, b, c…) y cada partición se denomina por un número (1, 2, 3…). Los discos llevan el prefijo sd.

Diagrama

Descripción generada automáticamentePor ejemplo, sda1, sda2, sda3…

Cada una de las particiones se puede usar para una función, por ejemplo, utilizar una para /home, otra para /var, otra para SWAP…

Este particionado es importante porque:

* Mejora la organización.
* Aumenta la seguridad (un problema en una partición no implica que se propague a otras particiones).
* Impide la pérdida de datos en caso de corrupción del sistema.

# VENTAJAS DE MONTAJE EN VARIAS PARTICIONES

* **Soporte multi-operativo** 🡪 Permite alojar varios SOs.
* **Elección del sistema de ficheros** 🡪 Cada partición puede tener su sistema de ficheros.
* **Control y administración del espacio en disco** 🡪 Control de acceso de los usuarios a nivel de particiones.
* **Protección de errores en el disco** 🡪 Si falla una partición no crítica el sistema puede seguir trabajando.
* **Seguridad** 🡪 Se puede montar el acceso a una participación en solo lectura.
* **Backup** 🡪 Las herramientas de copias de seguridad trabajan mejor con particiones pequeñas.

# INODOS

Los inodos son estructuras de datos que almacenan información sobre un archivo de nuestro sistema de archivos (es decir, metadatos de un archivo). No tienen nombre y se identifican mediante un número entero único y cada uno de ellos sólo puede contener los datos de un único archivo del sistema de archivos.

Los metadatos almacenados son:

* **Número de inodo**. Sirve para identificar al inodo.
* **Tamaño del archivo** y número de bloques que ocupa.
* **Dispositivo** **de almacenamiento** del archivo.
* **Número de enlaces**.
* **Identificador de usuario** (UID o UserID).
* **Identificador de grupo** (GID o GroupID).
* **Marcas de tiempo** (por ejemplo, la fecha de creación, la fecha de último acceso…).

Para acceder a estos datos utilizamos el comando stat seguido del nombre del archivo, por ejemplo:

Texto

Descripción generada automáticamente

Su función principal es el indicar las posiciones del disco duro en las que están almacenados los datos del propio archivo.

# ENLACES DUROS Y ENLACES BLANDOS

En Linux tenemos dos tipos de enlaces los cuales funcionan como los accesos directos de Windows: los enlaces duros, fuertes o *hard links* y los enlaces blandos, simbólicos o *soft links*.

* Los **enlaces duros** son archivos que apuntan al mismo contenido almacenado en disco que el archivo original, por tanto, archivos originales y enlaces duros disponen del mismo inodo y a consecuencia de ello ambos estarán apuntando hacia el mismo contenido almacenado en el disco duro.

Por tanto, un enlace duro es la forma de identificar en el disco duro un contenido con un nombre distinto al del archivo original.

Estos enlaces se podrán realizar siempre y cuando el archivo esté en la misma partición del disco que pretendemos crear el enlace.

Se utilizan para:

* + Realizar copias de seguridad incrementales ahorrando espacio en disco duro y un tiempo considerable.
  + Cuando copiamos un archivo de gran tamaño.
  + El enlace duro es una muy buena opción para tener un archivo en varias ubicaciones. Usando enlaces duros para este fin evita que se generen enlaces simbólicos rotos.
  + Los enlaces duros también nos pueden servir para clasificar información como por ejemplo fotografías.
* Los **enlaces blandos** son los más parecidos a los accesos directos de Windows. Apuntan al nombre de un archivo (un enlace duro) y, posteriormente, el archivo apunta a un contenido almacenado en nuestro disco duro.

Cada uno de ellos dispone de su propio número de inodo y es diferente al del archivo original.

Estos enlaces pueden ser creados, aunque estén en discos duros diferentes o en particiones diferentes.

Se pueden utilizar para:

* + Expandir sistemas de archivos.
  + Conseguir que un archivo esté disponible en 2 ubicaciones diferentes de forma rápida y sencilla.
  + Evitar tener que copiar directamente un archivo, lo que disminuye el tiempo de espera y se ahorra espacio en disco.
  + Sincronizar archivos y carpetas en Dropbox sin que el archivo que queremos subir a Dropbox esté dentro de la carpeta de Dropbox.