



# Gómez Miranda Leopoldo

## Tarea: Tipos de Archivos UNIX

En el sistema UNIX hay cuatro tipos de ficheros: ordinarios (también llamados regulares o de datos), directorios, ficheros de dispositivos (conocidos también como ficheros especiales) y tuberías (en inglés pipes o FIFOs)

### Ficheros ordinarios.

Los ficheros ordinarios contienen bytes de datos organizados como un array lineal. Las operaciones que se pueden hacer con los datos de un fichero son:

- Leer o escribir cualquier byte del fichero.
- Añadir bytes al final del fichero, con lo que aumenta su tamaño.
- Truncar el tamaño de un fichero a cero bytes. Esto es como si borrásemos el contenido del fichero.

Se debe tener presente que las siguientes operaciones no están permitidas con los ficheros:

- Insertar bytes en un fichero, excepto al final. No hay que confundir la inserción de bytes con la modificación de los que ya existen (proceso de escritura).
- Borrar bytes de un fichero. No hay que confundir el borrado de bytes con la puesta a cero de los que ya existen.
- Truncar el tamaño de un fichero a un valor distinto de cero.

### Directorios.

Los directorios son los ficheros que nos permiten darle una estructura jerárquica a los sistemas de ficheros UNIX. Su función fundamental consiste en establecer la relación que existe entre el nombre de un fichero y su nodo-i correspondiente.

El tamaño de cada entrada del directorio es de 16 bytes; dos dedicados al nodo-i y 14 dedicados al nombre del fichero. En la siguiente figura podemos ver la estructura típica de un directorio.

desplazamiento	nodo-i	Nombre del fichero
0	45	.
16	2	..
32	2.395	Inittab
48	371	Mount
64	4.921	shutdown

Ejemplo de estructura del directorio /etc para el UNIX System V.

Comando	Sintaxis	Descripción
mkdir	mkdir nom_directorio	Crea un nuevo directorio.



Los ficheros especiales, o ficheros de dispositivo, se utilizan para que los procesos se comuniquen con los dispositivos periféricos: discos, cintas, impresoras, terminales, redes, etc.

Hay dos familias de ficheros de dispositivos: ficheros modo bloque y ficheros modo carácter. Estos ficheros se crean mediante la llamada al sistema **mknod**

### Carácter

Los ficheros de dispositivo modo carácter la información no se organiza según una estructura concreta y es vista por el núcleo, o por el usuario, como una secuencia lineal de bytes. En la transferencia de datos entre el núcleo y el dispositivo no participa la memoria intermedia y por lo tanto se realiza a menor velocidad.

### Bloque

Los ficheros de dispositivo modo bloque se ajustan a un modelo concreto: el dispositivo contiene un array de bloques de tamaño fijo (generalmente múltiplo de 512 bytes) y el núcleo gestiona una memoria intermedia (buffer caché) que acelera la velocidad de transferencia de datos.

### Liga

Una liga no es una clase de archivo, sino un segundo nombre para un archivo. Si dos usuarios necesitan compartir la información de un archivo, ellos pueden tener copias separadas de este archivo. Un problema al mantener copias separadas es que las dos copias pueden rápidamente perder la consistencia. Por ejemplo, un usuario puede realizar cambios que el otro podría no conocer. Una liga proporciona la solución a este problema. Con una liga, dos usuarios pueden compartir un único archivo. Ambos usuarios parecen tener copias del archivo, pero solamente existe un archivo con dos nombres. Los cambios que cualquier usuario realiza tienen lugar sobre la versión común. Esta liga no solamente ahorra espacio al tener una única copia de un archivo, sino que asegura que la copia que cada uno utiliza es la misma. A este tipo de vínculo también se le denomina liga dura (hard links). Una liga dura se crea con **ln** y se borra con **rm**.

### Socket

Los "sockets" o "enchufes" permiten la comunicación entre procesos, muchas veces a través de la red. Los sockets de dominio son propios de una máquina; se los referencia como objetos de un sistema de archivos y no como puertos de red. Los archivos "socket" sólo pueden ser leídos o escritos por los procesos involucrados en la comunicación, aunque son visibles como entradas de directorio. Los sockets de dominio se crean con la llamada al sistema **socket()**, se eliminan con **rm**, o con la llamada al sistema **unlink()** si ya no tienen usuarios.

### Tubería

Una tubería es un fichero con una estructura similar a la de un fichero ordinario. La diferencia principal con estos es que los datos de una tubería son transitorios. Esto quiere decir que los datos desaparecen de la tubería a medida que son leídos. Las tuberías se utilizan para comunicar procesos.



Los datos escritos en la tubería se leen en el mismo orden en el que fueron escritos, siguiendo la disciplina de una hilera: el primer dato en entrar es el primero en salir. La sincronización del acceso a la tubería es algo de lo que se encarga el núcleo.