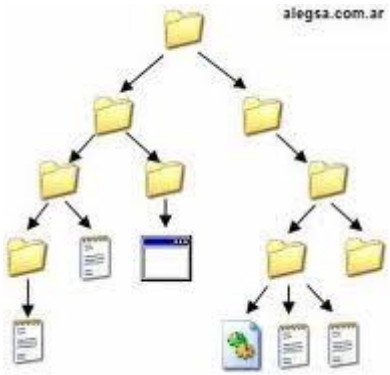
IES Rodrigo Caro

Dpto de Informática

7.- Ficheros y directorios.

Índice:



1.- Introducción.

En este tema veremos los comandos básicos para manejar ficheros y directorios.

¿De dónde partimos?.

Cuando entramos en el sistema, nos asignan un directorio por defecto.

Ejemplo:

Usuario → /home/usuario

Heidi → /home/heidi

Existe una variable llamada HOME, que es la que guarda ese directorio por defecto. $ echo $HOME

Podemos utilizar el comando pwd para averiguar en qué directorio de trabajo nos encontramos.

2.- Comandos para manipular ficheros y directorios.

2.1.- Comando cd

Sirve para acceder a un directorio.

$ cd [directorio]

$ cd	→	Nos lleva al directorio de trabajo.
$ cd ..	→	Accedemos al directorio padre.
$ cd . Ó cd ./	→	Accedemos al propio directorio.

2.- Comandos para manipular ficheros y directorios.

2.2.- Comando ls

Nos permite visualizar el contenido de un directorio.

$ls [opciones] [directorio|fichero]

$ls –l → Muestra un listado largo: Permisos, Nº enlaces, …….

$ls –a → Muestra todos los archivos, incluidos los ocultos.

Los archivos ocultos empiezan por .

$ls –R → Listado recursivo.

$ls –d → Muestra los nombres de los directorios como otros archivos en vez de listar su contenido.

2.- Comandos para manipular ficheros y directorios.

2.3.- Comando mkdir

Sirve para crear directorios.

$mkdir [opciones] directorio

Opción:

-p → Nos permite crear directorios intermedios.

2.4.- Comando rmdir

Sirve para borrar directorios vacíos.

$rmdir [opciones] directorio

Opción:

-p → Nos permite borrar directorios intermedios, siempre que estén vacíos.

2.- Comandos para manipular ficheros y directorios.

2.5.- Comando rm

Permite borrar ficheros y directorios incluso llenos.

$rm [opciones] <fichero|directorio>

Opción:

- -r → Nos permite borrar directorios aunque estén llenos.
- -i → Nos pide confirmación de borrado.
- -f → Forzado, ignora archivos no existentes y elimina cualquier aviso de confirmación.

2.- Comandos para manipular ficheros y directorios.

2.5.- Comando rm

2.- Comandos para manipular ficheros y directorios.

2.6.- Comando cp

Permite copiar archivos y directorios.

$cp [opciones] origen destino

Opción:

-R → Nos permite copiar de forma recursiva. Es decir, si el directorio tiene información, copia el contenido del directorio indicado.

2.- Comandos para manipular ficheros y directorios.

2.7.- Comando mv

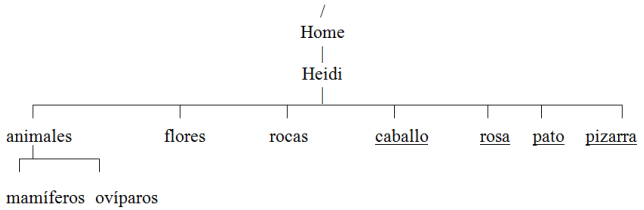
Permite mover o renombrar archivos o directorio.

$mv [opciones] origen destino

2.- Comandos para manipular ficheros y directorios.

2.6. y 2.7.- Comando mkdir, rm, cp y mv

Ejercicios: Previamente instala la herramienta tree. Aplícala para ver la estructura de directorios.



1.- Crea la siguiente estructura de directorios y archivos

Nota: Archivos → caballo, rosa, pato y pizarra.

2.- Desde el directorio de trabajo: /home/heidi

.-Mueve pato a ovíparos. .- Mueve pizarra a rocas. .- Mueve rosa a flores.

3.- Posiciónate en /home/heidi/animales/mamíferos y mueve el archivo caballo a mamíferos.

4.- Desde el directorio /home/heidi, crea un directorio que se llama copias.

5.- Realiza una copia de directorio animales dentro del directorio copias que has creado en el apartado anterior. → $cp -R animales/* copias $cp -R animales/ copias

6.- Copia el archivo rosa que está dentro de flores al directorio copias.

7.- Realiza un listado recursivo y borra la estructura anterior.

2.8.- Comando diff

El shell de linux: Comando diff

El comando diff nos permite comparar dos ficheros linea a linea y nos informa de las diferencias entre ambos ficheros. Diff tiene muchas opciones. Las que más uso son -w, -q, -y.

La sintaxis del comando es la siguiente:

diff [opciones] [fichero1] [fichero2]

Si queremos comparar dos ficheros, ignorando los espacios en blanco, utilizaremos el parámetro -w:

diff -w fichero1 fichero2

Si lo que queremos es que no nos muestre las diferencias, sino que tan sólo nos informe de si son diferentes o no:

diff -q fichero1 fichero2

Si queremos que nos muestre la salida con las diferencias marcadas a dos columnas:

diff -y fichero1 fichero2

Como en muchos otros comandos, también podemos utilizar la opción -i, que ignora la diferencia entre mayúsculas y minúsculas.

2.- Comandos para manipular ficheros y directorios.

2.9.- Comando wc y nl

$ wc archivo → Muestra la cantidad de líneas, palabras y bytes (caracteres) que contiene un archivo.

$nl archivo → Muestra el archivo numerado.

Ejercicio:

1.- Realiza un listado del directorio /

2.- Modifica el listado anterior para que nos muestre el número de archivos /directorios .

$ ls -l /|wc …..

3.-Metacaracteres

Son comodines que nos permiten seleccionar un conjunto de archivos.

*→ Representa a un conjunto de 0 ó más caracteres.

?→ Representa 1 carácter.

[ ]→ Representa caracteres individuales o un rango de caracteres:

- - → Rango
- ! → Excluir caracteres

3.-Metacaracteres

Ejercicios:

Accede al directorio /bin y anota el comando y el número de archivos mostrados después de realizar las siguientes secciones.

1.- Todos los archivos que empiecen por t.

2.- Todos los archivos que empiecen por b, terminen por h y tengan dos caracteres entre ambas letras.

3.- Todos los archivos que terminen en b o d o f o g.

4.- Todos los archivos que terminen en a o b o c o d.

5.- Todos los archivos que terminen en p.

6.- Todos los archivos que no terminen en p.

7.- Todos los archivos que empiecen por b, les siga un carácter comprendido entre u y z, y que termine por cualquier conjunto de caracteres.

8.- Todos los archivos que empiecen por b, les siga un carácter que no esté emprendido entre la u y la z y que termine por cualquier conjunto de caracteres.

4.-Enlaces

Sirven para referenciar a un fichero.

4.1.- Concepto de i-nodo.

Una definición simple es que se trata de un número entero que contiene toda la información necesaria para acceder a un archivo. Podemos verlo como un puntero al disco.

Tabla de directorios

15	Leeme
32	Presupuesto
32	presup

Un ejemplo de dos archivos que tienen el mismo i-nodo.

Tabla de inodos

Puntero	Propietario	Tipo de archivo	…….	Entradas bloque	Tamaño archivo
15				Bloq4, bloq 11	

El nombre del archivo no forma parte de la tabla de i-nodo.

Bloque de datos

1	2	3	4 00011111	5	6
7	8	9	10	11 10001011	12

Tamaño del bloque por defecto es 4096 bytes= 4KB

Cuando se formatea el disco se crea el espacio de inodos. Dependiendo del tamaño del disco duro habrá más o menos i-nodos.

4.-Enlaces

Sirven para referenciar a un fichero.

4.1.- Concepto de i-nodo.

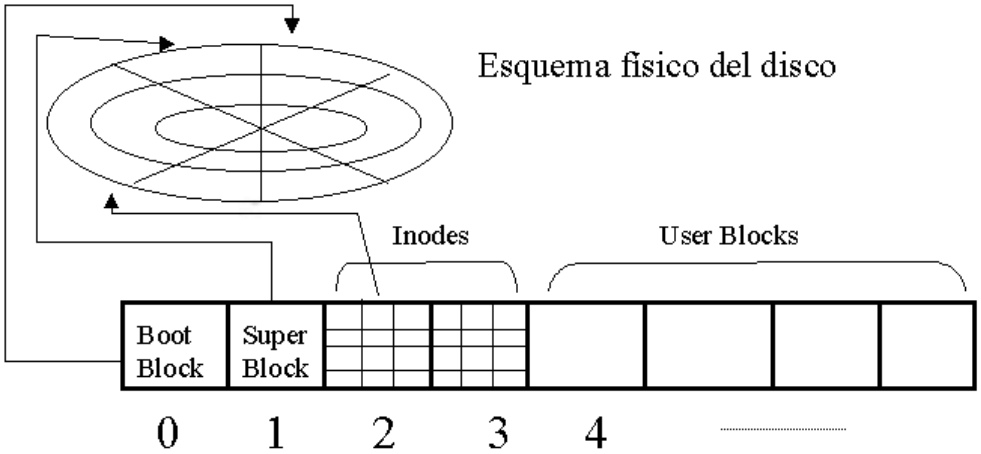
Un simil muy tonto

Tabla de directorios Lista de la clase	
15	Antonio
32	Juan José
32	Boni Es un apódo de Juan José

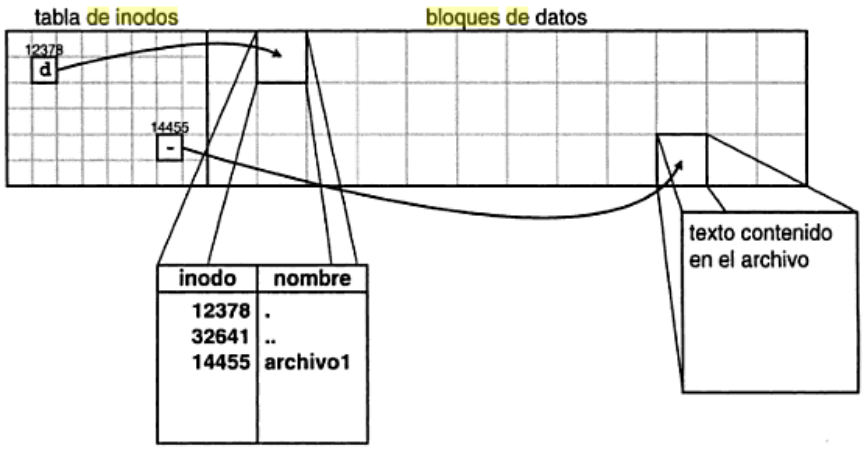
Tabla de inodos Información de los alumnos de la clase					
Puntero	Propietario	Tipo de archivo	…….	Entradas bloque	Tamaño archivo
15				Puesto 1 Puesto 2	

4.-Enlaces

4.1.- Concepto de i-nodo.



Esquema lógico del disco



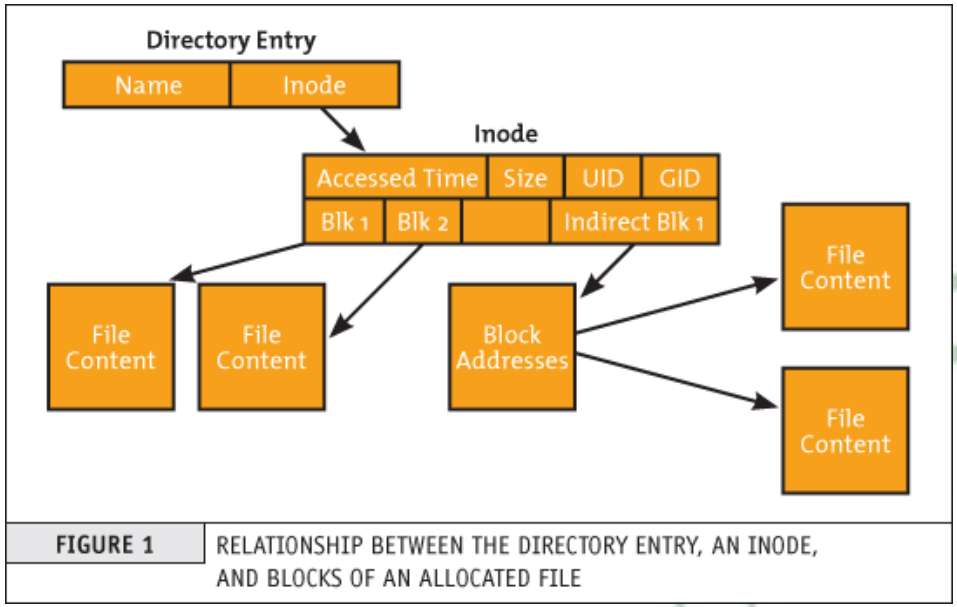


FIGURE 1 RELATIONSHIP BETWEEN THE DIRECTORY ENTRY, AN INODE, AND BLOCKS OF AN ALLOCATED FILE

4.-Enlaces

4.1.- Concepto de i-nodo.

Ejercicios:

1.- Crea dos archivos llamados nube y carta.

Ver los i-nodo asignados a cada uno de los archivos: $ls -i

4.-Enlaces

4.2.- Tipos de enlaces.

Podemos hablar de:

- Enlaces duros o físicos.
- Enlaces blandos o simbólicos.

4.3.- Enlaces duros o físicos

Si cambiamos el archivo referenciado de lugar o nombre, el enlace no se ve afectado.

Tienen el mismo i-nodo.

- $ln [opciones] origen [destino]

4.-Enlaces

4.4.- Enlaces blandos o simbólicos.

Serían equivalente a los accesos directos de Windows.

Si se cambia el archivo de nombre o se mueve, el enlace no funcionará.

Tienen distintos inodos.

Ventaja frente a los enlaces duros:

Se puede referenciar directorios.

```
$ ln –s origen [destino]
```

Nota: Indicar toda la ruta, tanto del origen como del destino.

4.-Enlaces

Ejercicios:

1.- Crea un archivo llamado carta.

2.- Realizar un listado largo de carta y apunta la línea obtenida.

```
-rw-r--r-- 1 usuario usuario 0 feb 10 06:36 carta
-rw-r--r-- 1 usuario usuario 0 feb 10 06:36 nube
```

3.- Crea un enlace duro a carta llamado cartita. Realiza un listado en formato largo e inodos.

```
171821 -rw-r--r-- 2 usuario usuario 0 feb 10 06:36 carta
171821 -rw-r--r-- 2 usuario usuario 0 feb 10 06:36 cartita
171822 -rw-r--r-- 1 usuario usuario 0 feb 10 06:36 nube
```

4.- Accede al archivo carta y escribe “ Ya se ha perdido la costumbre de escribir cartas”.

5.- Visualiza el contenido del archivo cartita.

6.- Cambiamos el nombre de carta y le ponemos email.

7.- ¿Podemos acceder a cartita?.

8.- Borra email y cartita.

4.-Enlaces

Ejercicios:

1.- Crea un archivo llamado nube.

2.- Realizar un listado largo de nube y apunta la línea obtenida.

```
-rw-r--r-- 1 usuario usuario 0 feb 10 06:45 nube
```

3.- Crea un enlace blando a nube llamado nubecita. Realiza un listado en formato largo e inodos.

```
171821 -rw-r--r-- 1 usuario usuario 0 feb 10 06:45 nube
171822 lrwxrwxrwx 1 usuario usuario 4 feb 10 06:46 nubecita -> nube
```

4.- Accede al archivo nube y escribe “Las nubes son gotas de agua suspendidas en la atmósfera”.

5.- Visualiza el contenido del archivo nubecita.

6.- Cambiamos el nombre de nube por lluvia.

7.- ¿Podemos visualizar el contenido del archivo nubecita?.

8.- Borra lluvia y nubecita.

5.- Localizar archivos.

Nos permite buscar archivos en la estructura de directorios.

$find [ruta] [criterio] [accion]

[ruta]: Es el lugar en el que empieza la búsqueda. Se hace de forma recursiva.

[criterio]: Condiciones de búsqueda:

-name “nombre archivo”
-user nombre_usuario
-iname “nombre archivo”

Si utilizamos comodines debe ir entre comillas.

[accion]: La acción que realizará sobre los archivos que cumplan la condición.

-exec comando {} continuación_comando \;

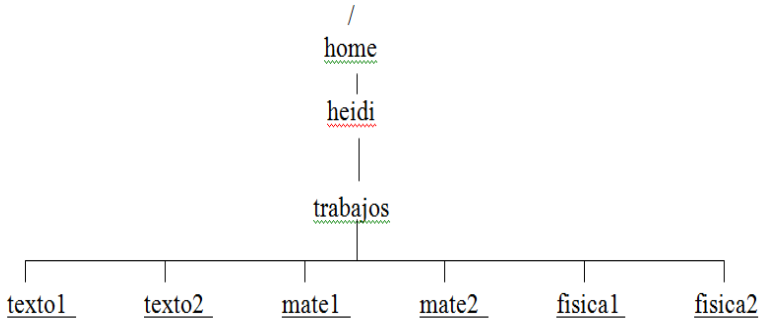
Ejemplo:

$ find / -iname “joe*” –exec cp {} /home/heidi \;

5.- Localizar archivos.

Ejercicios:

1.- Crea la siguiente estructura.



2.- Desde el directorio de Heidi:

A. Encontrar el archivo llamado fisica1
B. Busca todos los archivos que empiezan por fis
C. Busca todos los archivos que se encuentran en /usr del tipo txt.*
D. Busca todos los archivos del usuario Heidi desde la raíz que empiece por “mat”

3.- Crea una carpeta llamada apuntes en /home/heidi.

A. Encontrar todos los archivos dentro de la carpeta trabajos que empiecen por mates.
B. Encontrar todos los archivos dentro de la carpeta trabajos que empiecen por mates y moverlos a la carpeta apuntes.
C. Mover todos los archivos dentro de la carpeta trabajos que empiecen por “text” a la carpeta apuntes.