

Gestión de archivos en Linux

Gestión de archivos y almacenamiento

Tipos de archivos en Linux

- **Regulares:** son ficheros ordinarios que contienen información de diversa naturaleza. Los ficheros ejecutables son un tipo de fichero regular que tiene activo el permiso de ejecución y contiene código ejecutable.
- **Directorios:** Almacenan en su bloque de dtos el número de i-nodo y nombre de los archivos que contienen. Cuando son listados con el comando ls muestra la información accediendo a la estructura interna de cada i-nodo.

Tipos de archivos en Linux

- **Enlaces:**

- Enlaces duros: son asociaciones de nombres de ficheros a i-nodos. Se trata de reutilizar un i-nodo asignándoles nombres distintos y localizándose en diferentes directorios o en el mismo. Los enlaces duros se establecen sobre ficheros regulares dentro del mismo sistema de archivos
- Enlaces simbólicos o blandos: el directorio que contiene un enlace simbólico almacena un i-nodo (diferente al i-nodo al que enlaza) y un nombre de fichero. El i-nodo del enlace simbólico almacena la ruta al archivo definitivo. Se usa para referenciar a archivos de otras particiones y equipos en red

- **Dispositivos:** son archivos que representan a dispositivos físicos. La mayoría de ellos se encuentran en el directorio /dev. Hay dos tipos

- Dispositivos por caracteres: no disponen de un sistema de archivos propio (impresoras, teclados...)
- Dispositivos por bloques: almacenan la información en bloques de datos.

Comandos para gestionar archivos

- ls: lista ficheros y directorios
- cd: sirve para cambiar de directorio
- touch: crea un fichero.
- mkdir: crea carpetas
- ln: crea enlaces
- rm: elimina ficheros y directorios
- cp: copia archivos y directorios de una localización a otra.
- mv: mueve archivos y directorios de una localización a otra.
- cat: imprime en pantalla el contenido de un archivo
- pwd: imprime en pantalla el directorio actual

Redirecciones de la salida estándar y error

- Podemos emplear el operador `>` para volcar la salida estándar de una orden sobre un fichero. Si el fichero existiese se sobrescribiría con la salida estándar
- Si utilizamos el doble `>>`, la salida estándar se añadirá al fichero en caso de que este ya existiese.
- Además de la salida estándar existe la salida de errores que contendrá los mensajes de error que se han producido al ejecutar el script. Para redireccionar esta salida a un fichero haremos `2>` o `2>>`
- También se puede redireccionar a la salida estándar utilizando los operadores `1>` y `1>>`
- Para redireccionar ambas salidas al mismo fichero usaremos `&>` y `&>>`

Redirecciones de la salida estándar y error

- Cuando se quiere redirigir la salida de error y estándar a la propia salida estándar se utilizará la orden `2>&1`

```
invalid command > error.log 2>&1
```

- Si se desea descartar la salida de un proceso se puede utilizar el dispositivo virtual `/dev/null`, conocido como cubeta de bits.

```
echo Hola > /dev/null
```

Redirecciones de la entrada estándar

- También es posible redireccionar un fichero como entrada de una orden, para ellos emplearemos el operador <
orden < fichero
- Existe una redirección que permite introducir texto al usuario hasta que se encuentre una línea con el delimitador establecido
<<delimitador

```
mikel@mi-xps:~$ cat <<END
> capitulo1
> capitulo2
> END
capitulo1
capitulo2
```

Procesamiento de texto

- cat: imprime en pantalla el contenido de un archivo
- head: imprime las primeras líneas de un archivo
- tail: imprime las últimas líneas de un archivo
- grep: busca un texto en archivos
- cut: divide cada línea en varias columnas

Expresiones regulares básicas

Es posible especificar expresiones regulares básicas en los comandos grep para realizar búsquedas más complejas

Símbolo	Significado	Ejemplos
.	Cualquier carácter, excepto el carácter final de línea	Cas.
*	Cero o más repeticiones del carácter que le precede	C*
[lista]	Coincide con uno de los caracteres presentes en la lista	[aCgh]
[^lista]	No coincide con ninguno de los caracteres presentes en la lista	[^aCgh]
[rango]	Se pueden expresar rangos de caracteres	[0-9] [^0-9]
^	Comienzo de línea	&C
\$	Fin de línea	a\$