LINUX. USO DEL SISTEMA OPERATIVO.

1. ESTRUCTURA DEL SISTEMA DE ARCHIVOS

El sistema de archivos de Linux tiene una estructura jerárquica, donde todos los directorios cuelgan de uno, llamado **raíz**, que se representa por el carácter "**/**". Colgando de él podemos encontrar otros directorios, y así sucesivamente.

El conjunto de un directorio del que cuelgan otros directorios con ficheros y más subdirectorios se conoce como **árbol de subdirectorios**.

De los directorios más importantes destaca el directorio **home**, que contiene los subdirectorios personales de los usuarios del sistema.

Otros directorios son los siguientes:

/bin → contiene los programas que pueden ejecutar todos los usuarios del sistema. Son archivos binarios o ejecutables.

/boot → contiene los archivos de configuración de arranque del sistema.

> **/boot/grub** → contiene el gestor de arranque GRUB.

/dev → contiene los archivos del hardware del sistema.

/etc → contiene los archivos de configuración del sistema.

> **/etc/init.d** → en este directorio se encuentran todos los shell scripts que facilitan el inicio y cierre de los daemons (disk and execution monitor) o "demonios", que son un tipo especial de proceso que se ejecuta en segundo plano para dar algún tipo de servicio. Estos scripts suelen tener como argumentos stop, Stara y restart. Estos argumentos generalmente provienen de lo que se especifica en los directorios **/etc/rc[0-6].d**
>
> rc0.d | rc1.d | rc2.d | rc3.d | rc4.d | rc5.d | rc6.d : estos subdirectorios contienen enlaces simbólicos hacia los shell scripts del directorio /etc/init.d, dependiendo del nombre del enlace, si empieza por S o por K, el argumento que envían a estos scripts es stop o start.

/etc/network → este directorio contiene los scripts ifup y ifdown para habilitar y deshabilitar las interfaces de red del equipo.

/home → es el directorio que contiene los directorios personales de todos los usuarios del sistema. Además, para cada usuario, tendrá el directorio:

/home/usuario/Escritorio → contiene el escritorio del usuario (en ocasiones aparece como Desktop en lugar de Escritorio).

/lib → contiene las bibliotecas necesarias para que se ejecuten los programas que tenemos en /bin y /sbin.

/lost+found → aquí encontramos información y los procesos que se ejecutaban antes de una caída del sistema.

/media → contiene las unidades físicas que tenemos montadas o que se montan automáticamente, como discos duros, unidades de DVD, pendrives…

/mnt → directorio donde se montan sistemas de archivos de forma temporal, cuando sea necesario.

/opt → aquí podemos instalar aplicaciones que no vienen en los repositorios y que por tanto no se instalan automáticamente.

/proc → contiene archivos que reciben o envían información al núcleo. Los ficheros que contiene realmente residen en memoria. Estos ficheros nos permiten obtener información sobre los procesos en ejecución. Por ejemplo, el fichero **partitions** contiene información de las particiones del disco. El fichero **filesystems** contiene los sistemas de archivos del sistema.

/root → directorio personal del superusuario o administrador del sistema.

/sbin → contiene programas que solo puede ejecutar el superusuario.

/usr → contiene los programas de uso general para todos los usuarios.

/usr/games → almacena juegos.

/usr/X11R6 → programas de X-Windows (el servidor gráfico).
/usr/bin → programas ejecutables de uso general.
/usr/doc → documentación del sistema.
/usr/lib → bibliotecas de los programas de los usuarios.
/usr/sbin → contiene los programas de administración del sistema.
/usr/src → almacena los códigos fuente de los programas.
/usr/share/doc, **/usr/share/man**, **/usr/share/info** → contienen la ayuda de cada aplicación instalada.

/tmp → contiene archivos temporales.

/var → contiene información variable, registros, datos de los servidores, etc.

/var/log → contiene los archivos de registro (logs).
/var/spool → contiene los eventos que se hayan programado con los comandos **at** y **con**.

2. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

Un archivo es un conjunto de información relativa a un mismo concepto y que se guarda bajo un nombre que lo identifica.

El nombre de un fichero debe tener entre 1 y 255 caracteres, que pueden ser cualquier carácter excepto el carácter "/" que se utiliza para el directorio raíz y para separar los nombres de directorios y ficheros cuando se escribe la ruta.

Se llama **i-nodo** (i-node o nodo índice) a cada entrada de una tabla con una estructura de datos, en la que se guarda información de cada fichero, llamada tabla de i-nodos. Esta tabla se crea al arrancar el sistema.

El sistema operativo identifica a cada fichero, además de por su nombre, por su número de i-nodo que es un número entero único para cada fichero dentro de todo el sistema de archivos.

Cada entrada en la tabla de i-nodos contiene la siguiente información sobre cada fichero:

- número de i-nodo.
- tipo de fichero.
- propietario y grupo.
- permisos del fichero.
- fecha de creación, acceso y modificación.

- vínculos o enlaces.

Cada vez que se cambie cualquier atributo del fichero, se actualiza la entrada del fichero en la tabla de i-nodos almacenada en memoria. Hay una copia de esta tabla en el disco que se actualiza regularmente. Los ficheros que no están en uso en ese momento tienen su i-nodo en el disco.

Un fichero puede ser de distinto tipo dependiendo de la función que realicen o de la información que contenga.

Dependiendo de la función, tenemos por un lado los ficheros binarios o ejecutables que contienen código interpretable por el ordenador, como las aplicaciones y los comandos, aunque en Linux también existen ficheros de texto, llamados shell script, que contienen a su vez comandos. Por otro lado, tenemos los ficheros que almacenan información y que para trabajar con ellos necesitamos los ficheros binarios o ejecutables.

2.1 ARCHIVOS O FICHEROS

Operaciones: crear, borrar, copiar, mover, modificar, seleccionar, renombrar, enlazar, abrir y cerrar.

ENLACES

Dentro de los ficheros podemos distinguir entre fichero regular, enlace (que es como otro nombre del fichero y que tiene el mismo número de i-nodo) y enlace simbólico, que es una especie de acceso directo a otro fichero o a un directorio.

Los enlaces se utilizan para crear otro nombre a un fichero o directorio en otra ubicación. Existen dos tipos de enlaces, enlaces duros y enlaces simbólicos.

Enlace duro. Si creamos un enlace duro a un fichero, los cambios realizados en el fichero enlace se reflejarán en el original y viceversa. Los dos ficheros tienen el mismo i-nodo, por lo que hacen referencia a la misma zona del disco. Es como un fichero que tuviera dos nombres. Si se borra uno de los dos ficheros, el otro sigue existiendo con su información intacta. Un mismo fichero puede tener más de un enlace.

Enlace simbólico. Es un acceso directo y rápido a un fichero desde otra ubicación. En este caso, cada fichero tiene su propio número de

i-nodo, porque el enlace simbólico lo que guarda es información en dónde está el fichero origen, que en el caso de que se borrara, el enlace simbólico perdería toda la información.

.
.
.

2.2 GESTIÓN DE ARCHIVOS Y DIRECTORIOS EN MODO TEXTO

En modo texto podemos realizar las mismas operaciones para los ficheros y directorios que hemos visto en el entorno gráfico. Antes de ver los comandos, vamos a ver unos conceptos necesarios para trabajar en modo texto.

Caracteres comodines

Los caracteres comodines se utilizan para sustituir a un carácter o a un conjunto de caracteres.
Los caracteres comodines que podemos usar son:

Carácter	Función
*	Hace referencia a una cadena de caracteres de cualquier tamaño, incluso de tamaño 0.
?	Hace referencia a un carácter.
[ ]	Hace referencia a un carácter. Dentro de los corchetes podemos incluir un conjunto de caracteres o un rango de caracteres, pero el corchete se sustituirá solo por un carácter.
{ }	Hace referencia a varias cadenas de caracteres, que se escribirán dentro de las llaves separadas por comas.

Ejemplos:
acme* → √ acme ; √ acme2 ; √ acme23a ; X acm
*acme → √ aacme ; √ acme ; X acmed
acme → √ acme ; √ aacmed ; X bcme
smartphone?? → √ smartphone92 ; √ smartphoneAB ; X smartphoneABC

logo[ABC] → √ logoA ; √ logoB ; √ logoC ; X logoABC ; X logoAB ; X logoD
logo[!ABC] → √ logoD ; √ logo4 ; X logoA ; X logoB ; X logoC ; X logoDL
logo[?F] → X logo3F ; X logoFF ; √ logoF ; X logoFFF ; √ logo4

Caracteres de escape
Hay ciertos caracteres que no son imprimibles y otros, como en el caso del carácter espacio a la hora de utilizarlo en la línea de comandos, podríamos tener problemas porque el sistema puede creer que estamos introduciendo dos o más argumentos en vez de un argumento que tenga espacios en blanco. En este caso, para el nombre de archivos y directorios se protege el nombre poniéndolo entre comillas, o bien utilizando el carácter de escape \

Comandos

ls (list)
muestra información sobre ficheros y directorios. Si no se especifica nada, muestra información sobre el directorio actual.

Sintaxis: ls [opciones] [argumentos]

Opciones:
-d → --directory ; muestra información sobre el directorio en vez de sobre el contenido del directorio.
-a → --all ; permite ver los nombres de ficheros y directorios que comienzan por un punto, es decir, los ocultos.
-A → --almost-all ; permite ver los nombres de los ficheros y directorios que empiezan por un punto excepto los directorios . y ..
-l → muestra la información en formato largo, con información adicional, como tipo de archivo, tamaño, fecha de modificación, propietario y permisos.
-h → --human-readable ; junto con "l" o "s" muestra el tamaño en la unidad de medida mayor, para que se pueda entender mejor.
-i → --inode ; muestra el número de i-nodo del fichero.
-n → --numeric-uid-gid ; igual que "l" pero mostrando el número GID y UID en lugar de los nombres de usuario y grupo.
-c → muestra la información ordenada por día y hora de creación.
-t → el orden es por día y hora de modificación.

-r → --reverse ; muestra el resultado ordenado por orden inverso.
--color → muestra el contenido coloreado.
-F → --classify ; muestra información sobre el tipo de fichero. Los símbolos que aparecen junto al nombre del fichero indican * ejecutable, / directorio, @ enlace simbólico (o → si lo usamos con la opción -l), | tubería, ningún símbolo indica fichero regular.
-R → --recursive ; muestra los directorios por debajo del actual de forma recursiva.
-s → --size ; muestra el tamaño en bloques de cada fichero.
-S → muestra los ficheros ordenados por tamaño.

Ejemplos:

a) Desde tu directorio personal, muestra en formato largo los ficheros tty0 a tty9 del directorio /dev, utilizando caracteres comodines.

b) Muestra ahora los ficheros del directorio /dev que empiecen por tty, sin importar qué caracteres haya después.

c) muestra las entradas de tu directorio personal, de manera que te muestre el i-nodo de cada una e información sobre el tipo de fichero.

d) muestra las entradas de tu directorio personal, en formato largo

e) muéstralo ahora con el tamaño de los ficheros en la unidad de medida mayor que se pueda

Solución:

a) ls -l /dev/tty[0-9] o bien ls -l /dev/tty{0,1,2,3,4,5,6,7,8,9}
b) ls -l /dev/tty*
c) ls -Fi
d) ls -l
e) ls -lh

Ahora que ya conoces el funcionamiento del comando ls, sería útil si buscas información del significado de cada uno de los campos obtenidos con la opción -l.

pwd (print working directory)

print working directory. Muestra la ruta absoluta del directorio donde estamos en ese momento.

mkdir (make directory)

crea directorios

cd (change directory)

cambia de directorio.

Opciones:
- → cambia al último directorio en el que estuvimos antes de estar en el actual.
.. → cambia al directorio por encima del actual (directorio padre).
~ → cambia al directorio personal del usuario (home). También se puede acceder al mismo si no escribimos ninguna opción.

Argumento:
Cambia al directorio que se especifique como argumento.

Ejemplos:

a) Muestra la ruta absoluta del directorio donde estás. Muévete al directorio raíz. Ve ahora al directorio /etc/init.d Utiliza el comando que te lleve al directorio de donde vienes. Comprueba dónde estás.

b) Ve a tu directorio personal. Comprueba que los directorios "." y ".." son enlaces duros al mismo subdirectorio y al directorio padre.

Solución:

a)
pwd
cd /
cd /etc/init.d
cd -
pwd

b)
cd (o bien cd ~)
ls -ai (y miramos el i-nodo de . y ..)
cd ..
ls -di /home (comprobamos el i-nodo del directorio /home)
ls -i (comprobamos el i-nodo del directorio personal)
Son enlaces duros porque el número de i-nodo del directorio . coincide con el directorio personal, y el número de i-nodo de .. coincide con el del directorio /home

rmdir (remove directory)

borra los directorios si están vacíos

rm (remove)

borra ficheros y directorios

Opciones:
-f → descarta los ficheros que no existan, sin preguntar.
-i → pregunta antes de borrar cada fichero o directorio.
-r → -R → borra los directorios, los ficheros que contengan y los subdirectorios, de forma recursiva.
-v → muestra un mensaje por cada directorio o fichero borrado.

cp (copy)

copia uno o varios ficheros en otro fichero o en un directorio.

Opciones:
-a → copia de forma recursiva manteniendo los permisos
-f → fuerza la copia. Si el destino existe y no se puede abrir, lo borra e intenta copiar de nuevo.
-i → pregunta antes de sobrescribir.
-R → copia directorios y los que están por debajo de él, de forma recursiva.

mv (move)

mueve un fichero o ficheros a otro fichero o directorio. Es equivalente a una copia seguida del borrado del original. Puede ser usado para renombrar ficheros.

Opciones:
-u → mueve solo si el destino no existe o es anterior al fichero fuente.
-i → pregunta antes de sobrescribir.
-f → fuerza la sobrescritura.
-v → muestra un mensaje por cada fichero movido.

file

muestra el tipo de fichero

du (disk usage)

muestra el espacio que ocupa el fichero o directorio

Opciones:
-b → muestra el tamaño en bytes
-h → muestra el tamaño en la unidad de medida mayor, para que se entienda mejor

df (display free)

muestra el espacio libre en los dispositivos de almacenamiento

Opciones:
-h → muestra el tamaño en la unidad de medida mayor
-k → --blok-size=1K
-a → muestra todos, incluso los que tengan tamaño 0

cat (catenate)

muestra el contenido de los ficheros que se le pasen como argumentos.

Opciones:
-n → enumera todas las líneas.

head

muestra las 10 primeras líneas de los ficheros que se indiquen

Opciones:

-n → muestra las n primeras líneas en lugar de las 10 primeras.

tail

muestra las 10 últimas líneas de los ficheros que se le indiquen

Opciones:
-n → muestra las n últimas líneas

wc

muestra el número de líneas, palabras, caracteres y bytes de los ficheros que se le indiquen, o el tamaño de la línea más larga.

Opciones:
-c → muestra el número de bytes
-m → muestra el número de caracteres
-l → muestra el número de líneas
-w → muestra el número de palabras
-L → muestra el tamaño de la línea más larga del fichero

more

muestra el contenido de los ficheros pero de forma paginada. A diferencia del comando cat, cuando muestre el contenido del fichero, si éste ocupa más de una pantalla, se quedará esperando que se pulse una tecla. Si es la barra espaciadora, se avanzará una página, y si es la tecla enter, se avanzará una línea. Con la tecla q finalizará la ejecución.

less

muestra el contenido de los ficheros de la misma forma que more, con la diferencia de que podemos movernos por ellos utilizando las teclas de cursor.

sort

muestra en orden ascendente el contenido de los ficheros que se le pasan como argumento. Además, lo podremos usar para concatenar ficheros de texto usando redireccionamientos y tuberías.

Opciones:
-c → comprueba que el fichero esté ordenado, pero no lo ordena. Si no está ordenado te muestra un mensaje indicando la primera línea que está fuera de orden.
-r → ordena en sentido inverso
-m → mezcla ficheros ya ordenados, no ordena
-u → elimina líneas repetidas
-n → ordena de forma numérica
-t → indica el separador de campos
-k? → donde ? es el número del campo (se usa en combinación con -t)

Debido a la cantidad de opciones de este comando, se aconseja consultar la ayuda: sort --help

ln

crea un enlace al fichero o directorio que se le especifique. Si es a un directorio, el enlace será simbólico.

Opciones:
-s → crea un enlace simbólico en vez de duro
-t → especifica el directorio donde se van a crear los enlaces

cut

muestra solo ciertas líneas de los ficheros que se le pasen como argumento.

Opciones:
-b → muestra solo los bytes que se le especifiquen
-c → muestra solo los caracteres que se le especifiquen
-d → usa el carácter que se le especifique como delimitador en vez del tabulador
-f → muestra sólo los campos que se le indiquen en la lista. Puede ser un campo, una serie de campos separados por comas o un rango.
-s → no muestra las líneas que no contengan el delimitador
--output-delimiter=cadena → usa la cadena como delimitador de salida en vez del delimitador de entrada.

grep

muestra las líneas de un fichero que coinciden con un patrón especificado.

Opciones:
-r → -R → para buscar de forma recursiva dentro de los ficheros de un directorio
-n → muestra el número correspondiente de la línea en que se encuentra el patrón
-i → no distingue entre mayúsculas y minúsculas
-v → muestra las líneas que no se corresponden con el patrón
-w → el patrón debe aparecer como una palabra completa y no como una parte de una palabra
-c → escribe el número de líneas que satisfacen la condición
-l → se escriben los nombres de los ficheros que contienen líneas buscadas

Patrón:
texto → líneas que contengan la cadena "texto"
^texto → líneas que empiecen por "texto"
^[^texto] → líneas que no empiezan por "texto"
texto$ → líneas que terminen por "texto"

tr

Traduce, reduce y/o borra caracteres de la entrada estándar, escribiendo en la salida estándar.

Sintaxis:
tr [OPCIÓN]... CONJUNTO1 [CONJUNTO2]

Opciones:
-c → -C → usa el complemento del conjunto1
-d → elimina el carácter o el grupo de caracteres indicados en conjunto1
-s → reduce a un solo carácter la repetición seguida de ese propio carácter, indicado como conjunto1
-t → traduce las coincidencias del conjunto1 y las convierte en el conjunto2

whereis

localiza los ficheros ejecutables o binarios, las fuentes y las páginas del manual correspondiente a los comandos o programas instalados que se pasen como argumento

which

muestra la ruta absoluta del archivo del comando o de los comandos que se le pasen como argumento

locate

busca archivos dentro del sistema de archivos. Solo puede hacer búsquedas por nombre de archivo. Es muy rápido porque busca en una base de datos propia que se va actualizando periódicamente.

find

busca ficheros en un árbol de directorios. Muestra el nombre de los archivos encontrados que se correspondan con cierto conjunto de criterios.

Opciones:
-follow → -L → sigue los enlaces simbólicos si apuntan a directorios

Criterios:
-type tipo → busca archivos de un tipo dado (f regular, d directorio, l enlace simbólico)
-name nombre → encuentra los archivos cuyo nombre coincida con el dado (-iname para que no distinga entre mayúsculas y minúsculas). Los criterios se pueden combinar con -a, -o, -not.
-maxdepth n → nivel máximo de subdirectorios a los que desciende buscando la información
-inum n → busca los ficheros que tengan el i-nodo n.

Acciones:
-exec comando → ejecuta un comando sobre cada archivo encontrado. La posición del archivo se indica con {} y el comando finaliza con ; que debe ser protegido mediante el carácter de escape \ para que el shell no lo interprete.

Ejemplos:

a) Busca en el directorio actual todos los ficheros con extensión doc o txt y muévelos al directorio /home/usuario/Documentos

Solución:

a)
find -name "*.doc" -o -name "*.txt" -type f -exec mv {} /home/usuario/Documentos \;

Filtros o tuberías

Las tuberías o filtros se utilizan en una línea de comandos para conectar la salida estándar de un comando con la entrada estándar de otro. Para ello se utiliza el carácter |

Ejemplos:

a) Muestra por pantalla el i-nodo de los ficheros de tu directorio personal, con el tipo de fichero que es y los permisos, y con el nombre del propietario del mismo. No tiene que aparecer ninguna información más, ni el nombre.

Solución:

a)
ls -li | cut -d " " -f1,2,4

Redireccionamientos

Cualquier proceso tiene una entrada estándar, **stdin**, y dos salidas, la salida estándar, **stdout**, y la salida de errores, **stderr**.

Normalmente, la entrada estándar, stdin, es el teclado y la salida estándar, stdout, es la pantalla, pero si se produjo un error en la ejecución del proceso, la salida por pantalla corresponderá a la salida de errores stderr.

Sin embargo, puede que queramos cambiar la entrada estándar, la salida estándar o la salida de errores por un fichero, por lo que habrá que usar los redireccionamientos, con los siguientes caracteres:

< → redirecciona la entrada estándar sustituyéndola por el archivo que se indique.

> → redirecciona la salida de un proceso al fichero que se le indique, borrando la información que el fichero contenía.

>> → redirecciona la salida de un proceso al fichero que se le indique, pero añadiendo esta salida al final del fichero, sin borrar lo anterior.

2> → redirecciona la salida de errores de un proceso al fichero que se le indique, borrando la información que el fichero contenía.

2>> → redirecciona la salida de errores de un proceso al fichero que se le indique, pero añadiendo esta salida al final del fichero, sin borrar lo anterior.

En todos los casos, si el fichero indicado no existe, se crea. El número 0 indica salida estándar, el número 1 indica salida estándar y el número 2 indica salida de error estándar, que es el único que es obligatorio escribir.

Con los redireccionamientos y los comandos cat y sort podemos crear y concatenar ficheros.

Comando relacionado con los redireccionamientos

Además de los caracteres vistos hasta ahora, existe un comando que también podemos utilizar para redireccionar la salida estándar hacia un fichero. La diferencia con el comando es que éste además de redireccionar al fichero, también muestra la información en la salida estándar, que normalmente es la pantalla.

tee

lee de la entrada estándar y escribe en la salida estándar y en un fichero que se le especifique.

Opciones:
-a → escribe al final del contenido del fichero, no lo sobrescribe.

-i → ignora señales de interrupción.

Ejemplos:

a) Crea un archivo llamado nuevo.txt con el comanto cat. Escribe en él varias palabras, una debajo de otra: zapato, cuchillo, perro.

Crea otro archivo, nuevord.txt con el comando sort. Escribe en él las mismas palabras que antes y una debajo de la otra.

Comprueba si nuevo.txt y nuevord.txt están ordenados.

Solución:

a)
cat > nuevo.txt
zapato
cuchillo
perro
CTRL+D

sort > nuevord.txt
zapato
cuchillo
perro
CTRL+D

sort -c nuevo.txt

sort -c nuevord.txt

Ejemplos:

b)
muestra el contenido del fichero raíz en formato largo y redirecciona la salida del comando a un fichero de tu directorio personal llamado inicio.txt

Intenta mostrar información de un fichero llamado ttt en el directorio raíz. Como no existe, mostrará error. Direcciona la salida de errores a un fichero llamado error.log

Solución:

b)

```
ls -l / > inicio.txt
ls /ttt
ls /ttt 2>error.log
```

3. ARCHIVOS ESPECIALES

Los archivos especiales tienen relación con las entradas y salidas (E/S). En Linux las entradas y salidas sobre un dispositivo se hacen mediante los archivos situados en ele directorio **/dev**. Cada uno de ellos se identifica por un nombre que indica de qué tipo de dispositivo se trata.

Existen varios tipos de archivos especiales. Si queremos ver de qué tipo de archivo se trata, podríamos mirar la salida del comando ls -l, donde el primer carácter de cada línea indica el tipo de archivo:

Carácter	Tipo de dispositivo	Función
c	de caracteres	Se utilizan para los dispositivos de E/S de caracteres como terminales, impresoras...
b	de bloques	Se utilizan para los dispositivos de bloques, como los discos.
s	sockets	Se utilizan para la comunicación de procesos a través de la red.
t	tuberías (pipes o fifo)	Se utilizan para comunicación entre procesos. Almacenan la información que se mandan entre sí.

En el directorio /dev encontramos la mayoría de los archivos especiales. Veremos los más utilizados, pero además de ellos hay muchos más y otros preparados para cuando se instalen nuevos dispositivos.

Dentro del directorio /dev tenemos archivos especiales de caracteres, tty, que hacen referencia a las terminales virtuales con las que podemos trabajar en el equipo. Tenemos 6 terminales virtuales en modo texto, **/dev/tty1** a **/dev/tty6** y una terminal en modo gráfico, /dev/tty7. Para cambiar de unos a otros tendremos que

pulsar las combinaciones de teclas CTRL+ALT+F1 a CTRL+ALT+F7, respectivamente, aunque existen más terminales en ele directorio por si las necesitamos instalar y utilizar.

Los archivos especiales de caracteres que hacen referencia a las pseudo terminales se encuentran en el directorio **/dev/pts**, con los nombres de 0,1,2,..., dependiendo de las que tengamos abiertas. Las pseudo terminales son procesos que emulan terminales en modo texto. En GNOME se pueden abrir mediante la aplicación Terminal de Aplicaciones o ejecutando desde una terminal abierta el comando **gnome-terminal**.

Dentro de los dispositivos de bloques, tenemos los que hacen referencia a los dispositivos de almacenamiento como los discos duros o los pendrives. **/dev/sda1** hace referencia a la primera partición del disco duro, dependiendo de lo que tengamos instalado. Si tenemos dos discos duros, los pendrives conctados se referenciarán con el nombre **/dev/sdc1**, **/dev/sdc2**...

Las unidades de DVD se referencian mediantes los archivos especiales de bloques **/dev/sr0**, **/dev/sr1**, dependiendo de lo que tengamos instalado.

Un dispositivo especial de caracteres es el dispositivo nulo **/dev/null**, que se utiliza para enviarle cualquier información que no queremos utilizar, por lo que al enviarla ahí, se perderá.

Ejemplos:

a)
Intenta ver el significado del primer carácter de la salida de ls -l, buscando ejemplos. Utiliza el comando file para ver la misma información, aunque de forma diferente.

Solución:

a)
el primer carácter de ls -l nos indica el tipo de fichero que es. Veamos los tipos posibles:

b → se trata de un dispositivo que se accede por bloques. Por ejemplo, un disco, el cual tiene acceso secuencial.
ls -l /dev/sda1
file /dev/sda1

c → dispositivo que se accede por caracteres. Por ejemplo /dev/null o una terminal.
ls -l /dev/null
file /dev/null

d → se trata de un directorio, por ejemplo, /home/usuario, /etc,...

l → indica que es un enlace simbólico.
ls -l /initrd.img
file /initrd.img

s → un socket o sistema de comunicación.
ls -l /dev/log
file /dev/log

p → indica que se trata de una tubería (pipe). Permite la comunicación entre procesos.
ls -l /dev/xconsole
file /dev/xconsole

- → es un fichero regular
ls -l texto.txt
file texto.txt

Ejemplos:

b)
si no sabes dónde hay un fichero de tipo socket, busca por todo el árbol de directorios, buscándolos desde el directorio raíz. Realiza lo mismo para los ficheros de tipo tubería pero evitando que los errores salgan por pantalla (redireccionándolos al dispositivo nulo).

Solución:

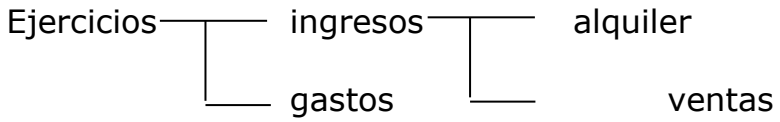
b)
ls -lR / | grep "^s"
ls -lR / 2>/dev/null | grep "^p"

EJERCICIOS

1. Utiliza un comando que te lleve a tu directorio personal directamente. Después ejecuta el comando que te dice la ruta

en la que estás y a continuación el comando que te dice quién eres.

2. Crea los siguientes directorios, desde el subdirectorio personal de usuario y sin salir de él.



3. Ve al directorio gastos. Utilizando el comando cat, crea dos ficheros emple1.txt y emple2.txt que contengan 3 palabras cada uno, una debajo de otra: roble, tigre, casa

4. Comprueba si emple1.txt está ordenado. Si no lo está, ordénalo y guarda el resultado en empleord.txt

5. Utilizando caracteres comodines, copia los 3 ficheros a ventas, sin salir de gastos y utilizando una trayectoria relativa.

6. Utilizando caracteres comodines, mueve los ficheros emple1.txt y emple2.txt a alquiler, sin salir de gastos y utilizando una trayectoria absoluta.

7. Ve al directorio raíz. Muestra el listado de todos los directorios en formato largo y después cambia la salida estándar a un fichero llamado directorios (que esté vacío) en tu carpeta personal.

8. Vuelve a gastos utilizando la opción del comando que te lleva al directorio anterior. Ve a alquiler utilizando una trayectoria relativa. Muestra el número de líneas del fichero emple1.txt. Utilizando el comando cat, une los ficheros emple1.txt y emple2.txt en emple3.txt. Muestra el número de palabras del fichero emple3.txt

9. Crea un enlace a emple2.txt llamado enlace2.txt. Modifica enlace2.txt. Comprueba que se ha modificado emple2.txt. Crea un enlace simbólico a emple3.txt llamado ensim3.txt. Muestra el listado largo de todas las entradas del directorio y mira el primer carácter de cada línea.

10. Borra los directorios y ficheros creados anteriormente.