

Uso básico del Interpretador de Comandos GNU/Linux

versión 1.0a

José Antonio Rey Morales



Índice de contenido

Ficheros y Sistema de Archivos GNU/Linux	2
Modo de Fichero	3
Parámetros y Sintaxis	4
Metacaracteres	5
Redireccionar salidas	5
Gestión de ficheros	6
Referencia rápida de comandos	11

1. Ficheros y Sistema de Archivos GNU/Linux

Para usar el interprete de comandos es necesario entender como funcionan los ficheros en GNU/Linux, y para entender como funcionan los ficheros es necesario entender su sistema de archivos.

La estructura interna del sistema de archivos en Linux es el siguiente:

Bloque de carga: En el se almacena el programa que gestiona el resto del sistema de archivos.

Superbloque: Contiene la información del sistema de archivos.

Tabla de i-nodos: Es la tabla que contiene los i-nodos.

→ **I-nodo:** Bloque que contiene metainformación sobre los ficheros y hace referencia a datos y a otras referencias a datos.

Zona de datos: Contiene los datos del usuario (Sistema operativo, archivos, directorios, etc).

En Linux, un fichero puede no tener ninguna extensión y ser usable.

Debemos tener muy en cuenta que Linux distingue entre mayúsculas y minúsculas (Case sensitive) por lo que Casa no es lo mismo que CASA o casa o cASa.

También tenemos que tener en cuenta que el nombre verdadero de un fichero es su ruta absoluta, y que en Linux los ficheros no se encuentran físicamente donde se indica, sino que se encuentran en la **tabla de "i-nodos"** y los ficheros hacen referencia los i-nodos de esta tabla.

Existen diferentes tipos de ficheros.

Archivo ordinario: Son archivos normales, como documentos de texto o programas.

Directorio: Son ficheros que hacen referencia a otros ficheros, ya sean archivos ordinario, otros directorios o ficheros especiales.

Vínculo físico: El vínculo físico es un tipo de archivo especial que apunta al mismo i-nodo del archivo original. El vínculo físico nunca puede ser un directorio porque podría crear bucles dentro del sistema de archivos.

Por ejemplo: Directorio_A y Vinculo_Fisico_Directorio_A

Si tuvieramos la siguiente ruta:

Directorio_A/Vinculo_Fisico_Directorio_A

Estaríamos dentro de un bucle porque Directorio_A haría referencia a su propio vínculo físico por lo que estaría haciendo referencia a sí mismo, que haría referencia a si mismo, ad infinitum.

Vínculo simbólico: A diferencia del vínculo físico, solamente hace referencia al fichero original y no a su i-nodo, por lo que pueden ser directorios.

Ficheros especiales: Ficheros carácter, bloque, cauce...

2. Modo de fichero

Los ficheros tienen una lista de permisos llamada "máscara de modo".

Esta máscara determina las acciones que se pueden realizar con el fichero en si, y se puede leer tanto en cadenas de caracteres como en octal.

Modos

Lectura: Permite visualizar el contenido del archivo o directorio.	Valor octal: 4
Escritura: Permite modificar el contenido del archivo o directorio.	Valor octal: 2
Ejecución: Permite ejecutar el archivo o entrar en el directorio.	Valor octal: 1

Para leer la máscara de modo en octal sumamos el valor de los permisos.

Lectura + Escritura + Ejecución = 7	Escritura + Ejecución = 3
Lectura + Escritura = 6	Escritura = 2
Lectura + Ejecución = 5	Ejecución = 1
Lectura = 4	Ningun permiso = 0

Deberemos leer la máscara de permisos en formato cadena de caracteres de la siguiente manera:

Lectura: r
Escritura: w
Ejecución x
Sin permiso: -

La máscara de modo se aplica sobre tres objetivos.

Dueño: El usuario dueño del fichero (Generalmente el usuario que lo creó).

Grupo del Fichero: El grupo al que pertenece el fichero (Generalmente el grupo principal del usuario que lo creó).

Otros: El resto de usuarios.

La máscara también indica que tipo de fichero se trata.

Archivo ordinario: - Vínculo: l
Directorio: d

También debemos leer la máscara en el orden preciso.

[Tipo de archivo] [Permisos dueño] [Permisos grupo] [Permisos otros]

Por ejemplo

```
-rwxr-xr--
```

```
754
```

Sería la máscara de permisos de un archivo ordinario, el dueño tendría todos los permisos, el grupo permisos de lectura y ejecución y el resto de lectura.

La máscara en octal no indica el tipo de archivo.

3. Parámetros y sintáxis

En el Shell utilizaremos de forma común una serie de parámetros para formar estructuras de ordenes completas para realizar las acciones que queremos.

Prompt: Es el indicador de la línea de comandos. Puede variar de un shell a otro. En Ubuntu, el prompt para usuarios corrientes es \$ y para el superusuario #.

Comando: Es la acción a realizar.

Opción: Las opciones modifican el comando para que realice las acciones de diferente forma, por ejemplo, que nos pida confirmación antes de borrar un archivo o nos ordene una lista de ficheros por orden cronológico.

Argumento: Son los objetivos del comando.

Generalmente, los comandos en la terminal de GNU/Linux suelen seguir la siguiente sintáxis:

```
usuario@equipo:~$ comando opciones argumentos
```

por ejemplo

```
jarey@laptop:~$ ls -lA /home/user/
```

El prompt es jarey@laptop:~\$

El comando es ls

Las opciones son l y A

El argumento es /home/user/

Se entendería de la siguiente manera:

Ejecutamos la orden ls de la manera l y A sobre el objetivo /home/user/

4. Metacaracteres

Los metacaracteres son caracteres especiales que funcionan de manera diferente al resto de caracteres.

Símbolo de cierre de interrogación: ?	Sustituye a un carácter cualquiera.
Asterísco: *	Sustituye a una cadena de caracteres de longitud indefinida.
Símbolo de cierre de exclamación: !	Excluye caracteres.
Corchetes: []	Se utilizan para incluir rangos de caracteres o para excluírlos si se usa junto al símbolo de cierre de exclamación.

Ejemplos

```
?asa      Casa, casa, tasa, Masa, pasa...
*asa      Arrasa, Car_casa, Mi_Casa...
[!a-c]asa tasa, Masa, pasa...
[Cc]asa   Casa, casa
t?[1-3]   ta1, ta2, ta3, tb1, tb2, tb3...
```

5. Redireccionamiento de salidas

Redireccionar salidas es una técnica que podemos utilizar para pasar la salida (El resultado de la ejecución) de un comando a un fichero o a otro comando.

Signo mayor que >	Es utilizado para pasar la salida de un comando a un archivo de texto.
Doble signo mayor que >>	Pasa la salida de un comando al final de un archivos de texto (Lo añade al final del documento).
Separación	Se utiliza para pasar la salida de un comando a otro.

Ejemplos

```
jarey@laptop:~$ ls -lA /dev/>/home/informe.txt
```

Este comando haría un listado en formato largo de todos los ficheros (Incluidos los ocultos, pero excluyendo . y ..) de la carpeta /dev/ de manera no recursiva (Solamente los ficheros de esa carpeta, no de subdirectorios) y pasaría el resultado de esta lista al archivo /home/informe.txt. Si el archivo /home/informe.txt no existe se crea en el momento. Si el archivo informe.txt tenía contenido se borra y solamente aparece el nuevo contenido.

```
jarey@laptop:~$ cat /etc/fstab | wc -l
```

Este comando primero recoge el contenido de /etc/fstab y después lo pasa a wc -l, el cual contaría el número de líneas de la salida de cat (El número de líneas del archivo /etc/fstab) y mostraría el resultado por la terminal.

5. Gestión de ficheros

Directorios

→ Crear un directorio

```
$ mkdir musica
```

→ Crear dos directorios con un comando

```
$ mkdir musica videos
```

→ Crear un directorio con espacios

```
$ mkdir "musica y videos"
```

→ Crear una jerarquía de directorios con un solo comando

```
$ mkdir musica musica/rock musica/grunge musica/grunge/nirvana
```

→ Crear una jerarquía de directorios recursivamente

```
$ mkdir -p musica/grunge/nirvana musica/rock/nickelback
```

→ Borrar un directorio vacío

```
$ rmdir musica/grunge/nirvana
```

→ Borrar una jerarquía de directorios (Un directorio con todo su contenido)

```
$ rm -r musica
```

Archivos

→ Crear un archivo

```
$ touch archivo
```

→ Visualizar el contenido de un archivo

```
$ cat archivo
```

→ Editar el contenido de un archivo, si el archivo no existe, se crea

```
$ nano archivo
```

→ Cambiar el nombre a un fichero

```
$ mv musica videos
```

→ Copiar un fichero

```
$ cp fichero_aqui fichero_alli
```

→ Copiar una jerarquía de ficheros (Un directorio y su contenido)

```
$ cp -r directorio
```

Empaquetado y compresión

→ Empaquetar ficheros

```
$ tar -cvf empaquetado.tar directorio
```

→ Desempaquetar ficheros

```
$ tar -xvf empaquetado.tar
```

→ Comprimir ficheros

```
$ tar -czvf empaquetado.tar.gz directorio
```

→ Descomprimir ficheros

```
$ tar -xzvf empaquetado.tar.gz
```

Información y navegación

→ Listar el contenido de un directorio

```
$ ls directorio
```

→ Cambiar el directorio actual

```
$ cd /etc
```

→ Cambiar al directorio home de nuestro login sin rutas

```
$ cd
```

→ Visualizar la máscara de permisos de un fichero

```
$ stat fichero
```

→ Visualiza la ruta completa del directorio actual

```
$ pwd
```

→ Visualiza el espacio usado en disco por el fichero y subdirectorios

```
$ du fichero
```

Propiedades de ficheros

→ Dar y quitar permisos a un fichero

```
$ chmod u+rwx,g+rw,o-wx fichero
```

→ Forzar una máscara de permisos concreta a todos

```
$ chmod a=rwx fichero
```

→ Forzar una máscara de permisos en octal

```
$ chmod 755 fichero
```

→ Cambiar el usuario dueño de un fichero

```
# chown nuevo_dueño fichero
```

→ Cambiar el grupo de un fichero

```
# chgrp nuevo_grupo fichero
```

Gestión de usuarios

→ Añadir un usuario, forzar la creación de su carpeta home, el shell bash y UID 1003
useradd -m -s /bin/bash -u 1003 usuario

→ Modificar el comentario de un usuario y añadirle a un grupo
usermod -c "Musica: Rock" -G rock usuario

→ Eliminar un usuario aunque este conectado
userdel -f user

→ Acceder al superusuario para un comando
sudo comando

→ Cambiar el usuario actual
\$ su usuario

→ Salir de la sesión del usuario actual
\$ exit

→ Mostrar el ID y los GID del usuario
\$ id usuario

→ Mostrar la información sobre los usuarios conectados
\$ who

→ Cambia la contraseña del usuario (Si no se pone usuario se obvia el actual)
passwd usuario

Gestión de grupos

→ Añadir un grupo con el GID 2012
groupadd rock -g 2012

→ Cambiar el nombre de un grupo
groupmod -n heavy rock

→ Eliminar un grupo
groupdel heavy

Gestión de Procesos

→ Visualizar los procesos de todos los usuarios en formato largo

```
$ ps -el
```

→ Visualizar los procesos en tiempo real

```
$ top
```

→ Visualizar los procesos en forma arborescente y mostrando los PID

```
$ pstree -p
```

→ Matar un proceso incondicionalmente

```
$ kill -9 PID
```

→ Mata una familia de procesos

```
$ killall nombreproceso
```

→ Inicia un proceso en segundo plano

```
$ proceso &
```

→ Pasa el proceso en primer plano actual a segundo plano

```
CTRL+Z
```

→ Muestra todos los procesos en segundo plano y su número de job

```
$ jobs
```

→ Pasa un proceso de segundo plano a primer plano

```
$ fg numero_job
```

→ Inicia un comando con -5 de prioridad

```
$ nice top -n -5
```

→ Cambia la prioridad de un proceso a -10

```
$ renice -10 PID
```

Gestión de servicios

→ Visualizar todos los servicios y ver su estado

```
$ service --status-all
```

→ Para el servicio de manera temporal

```
$ service servicio stop
```

→ Reanuda un servicio para esta sesión

```
$ service servicio start
```

→ Modifica los parámetros de un servicio para que se inicie con el sistema

```
$ update-rc.d servicio enable
```

→ Modifica un servicio para que no se inicie con el sistema

```
$ update-rc.d servicio disable
```

Montaje de dispositivos

→ Muestra una lista de los dispositivos montados

```
$ mount -l
```

→ Monta un dispositivo USB en /user/home/aqui_va con las siguientes opciones:

→ Escritura, Lectura, Ejecución

```
$ mount -t auto -o rw,exec /dev/sdb1 /user/home/aqui_va
```

→ Desmonta el dispositivo montado

```
$ umount /dev/sdb1
```

→ Configura el montaje del mismo dispositivo pero en fstab, añade la opción noauto

```
$ nano /etc/fstab
```

```
/dev/sdb1 /user/home/aqui_va auto rw,exec,noauto 0 0
```

Otros

→ Apaga el ordenador ahora

```
# shutdown now
```

→ Reinicia el ordenador

```
# shutdown -r
```

→ Visualiza una lista de las particiones

```
$ df
```

→ Busca información de un comando

```
$ man comando
```

→ Realiza una comprobación lógica del único disco duro IDE del equipo

```
# fsck /dev/hda
```

7. Referencia rápida de comandos

Gestión de usuarios y grupos

sudo	Permite identificarte como superusuario.
who	Muestra información sobre usuarios conectados.
passwd	Cambia la contraseña o las opciones de contraseña del argumento.
su	Cambia la sesión del usuario actual.
id	Muestra el UID y los GID del usuario.
exit	Sale de la sesión actual.
useradd	Crea un usuario nuevo.
usermod	Modifica los parámetros de un usuario existente.
userdel	Elimina un usuario existente.
groupadd	Crea un nuevo grupo.
groupmod	Modifica los parámetros de un grupo existente.
groupdel	Elimina un grupo existente.

Gestión de ficheros

cd	Cambia el directorio actual al argumento.
ls	Lista el contenido del argumento.
du	Calcula el espacio en disco usado por argumento.
pwd	Indica la posición actual del usuario.
mkdir	Crea un directorio con la ruta del argumento.
rmdir	Borra el directorio del argumento, no funciona si no está vacío.
touch	Actualiza la fecha de modificación del argumento, si el argumento no existe lo crea.
cp	Copia los ficheros del argumento.
mv	Cambia el nombre o mueve el fichero.
rm	Elimina el archivo argumento.
chmod	Modifica la máscara de modo del argumento.
chown	Modifica el dueño del argumento.
chgrp	Modifica el grupo del argumento.
umask	Cambia los permisos otorgados por defecto al crear ficheros.
stat	Muestra los permisos de los argumentos.
cat	Muestra el contenido de un archivo de texto plano.
nano	Editor de textos en terminal.
tar	Empaqueta, comprime o descomprime ficheros.
mount	Monta un dispositivo de almacenamiento al sistema de archivos.
umount	Desmonta el dispositivo de almacenamiento.
find	Busca los ficheros según los criterios establecidos.
locate	Busca los ficheros según los criterios, depende de una base de datos.
grep	Busca la cadena de caracteres en el archivo según los criterios.
wc	Cuenta el número de <opción> de los archivos especificados.

Gestión de procesos y servicios

ps	Muestra los procesos activos asociados al terminal.
top	Gestor de procesos en el terminal, se actualiza constantemente.
pstree	Muestra el árbol de procesos.
pidof	Muestra el PID del proceso.
service	Muestra el estado de los servicios y permite detenerlos o reanudarlos.
update-rc.d	Deshabilita o habilita el inicio de un servicio con el sistema.
bg	Manda el proceso de la terminal actual a segundo plano.
fg	Manda el proceso elegido a primer plano.
jobs	Visualiza una lista de los procesos de la terminal
nice	Ejecuta un comando con el nice indicado.
renice	Cambia el nice de los procesos al indicado.
kill	Permite enviar una señal para alterar el funcionamiento del proceso.
killall	Permite enviar una señal a toda la familia del proceso.

Otros

apt-get	Gestor de paquetes, instala y desinstala; actualiza paquetes y repositorios.
fsck	Comprueba la integridad lógica de un disco duro.
df	Visualiza un informe sobre el espacio usado en el disco.
man	Muestra el manual del comando.
tee	Permite redirigir la salida tanto a un archivo como a la terminal.
partimage	Lanza el gestor de copias de seguridad partimage desde la terminal.
uname	Muestra información del sistema.
shutdown	Apaga el equipo.