

Linux - Comandos básicos

Transformación a root

Para pasarnos a root simplemente haremos:

su -

Y colocaremos el password de root

Estructura básica de un comando

La estructura básica de los comandos esta dada por:

comand -[modificadores] parametro

Los modificadores lo que hacen es **modificar el comportamiento normal del comando**. Para **abortar** la ejecucion de un comando tenemos que apretar CTRL y sin soltar apretamos C.

Comando pwd

El primer comando a ver es “pwd” y me sirve para saber el path en el cual estoy parado.; pero no lo confundamos con el prompt.

\$ pwd

Change directory o CD

Este comando es el “change directory” que sirve para navegar por todo el árbol de directorios. Se puede utilizar tanto el path absoluto que se refiere a la posición exacta adentro del árbol de directorios:

```
$ cd /etc/defaults
```

Si yo coloco:

```
cd ..
```

Me estoy refiriendo al directorio inmediato superior.

Luego tenemos lo que son los path absolutos con los cuales referencio el path absoluto iniciando en la raíz del árbol de directorios.

```
$ cd /var/log
```

Listar archivos o ls

El ls lista los files de un directorio. Si yo lo ejecuto sin modificadores me lista únicamente los archivos (recuerden que para Linux **todo** es un archivo)

```
$ ls
```

El ls con el **-l** lo que hace es list long y nos va a mostrar cierta información que contiene el inodo:

```
$ ls -l
```

También hay una manera en la cual me va a mostrar con colores el resultado, dependiendo del tipo de archivo que sea y los permisos q tenga:

```
$ ls --color
```

También podemos usar path absoluto:

```
$ ls --color /var/log
```

O path relativo:

```
$ cd /var
```

```
$ ls --color -l log
```

Pero qué pasa si yo quiero que cuando ejecuto **ls** sea lo mismo que decir **ls --color**, utilizo algo llamado "**alias**".

```
$ alias ls='ls --color'
```

Entonces ejecuto:

```
$ ls -l log
```

Y llegamos al mismo resultado. Si por otro lado quiero ordenar por fecha de modificación de mayor a menor:

```
$ ls -lat /var/log
```

Pero si quisiera ordenarlo de menor a mayor agrego el "r".

```
ls -latr /var/log
```

Si yo quisiera ver el tamaño en la máxima unidad posible hago:

```
ls -lh /var/log
```

Y veo lo que me muestra el long, y aparte los archivos en K, M, o G.

Uso de disco o du

Este comando sirve para ver el Disk Usage y me va a mostrar cuanto espacio está ocupando un archivo, una carpeta, etc. Si yo pongo:

```
$ du /var
```

Me muestra el tamaño que ocupa cada carpeta y archivo desde donde yo estoy mostrando. Me lo muestra en bloque de 1K. Pero esto tiene un problema. Si yo listo por K, un archivo de 1K, me lo muestra de tamaños de 1M. Entonces volvemos al modificador **-h**

```
$ du -h /var
```

Entonces lo muestra en el bloque correcto. O lo que se conoce como human readable.

Espacio por filesystem df

El df me sirve para ver el espacio libre de cada filesystem. Si quiero ver de todos los filesystem o partición hago:

```
df
```

Si lo hago con el modificador **-h** lo que hace es mostrármelo en formato legible. Este “formato legible” va a seguir la misma logica en la que muestre tamaño:

```
df -h
```

History

El comando history me sirve para mostrar el historial de comandos ejecutados.

```
$ history
```

Made directory o mkdir

El comando mkdir sirve para crear directorios. Es lo mismo que decir “make directory”: por lo que vamos a crear una carpeta:

```
$ cd
```

```
$ mkdir A
```

Vamos a ver este mismo ejemplo con un path absoluto:

```
$ mkdir /root/A/B
```

Puedo crear varios directorios en un solo movimiento:

```
$ mkdir /root/LUNES /root/UNO
```

Pero lo que no puedo hacer por defecto es saltarme niveles en el árbol de directorios:

```
$ mkdir /root/A/B/C/D/E/
```

En este caso usamos la opción -p

```
$ mkdir -p /root/A/B/C/D/E/
```

RMDIR

Este es un comando para borrar directorios que se encuentren vacíos. Por eso es que no podemos borrar:

```
$ rmdir B
```

Pero si podemos borrar

```
$ rmdir /root/A/B/C/D/E
```

Mover o renombrar un archivo o directorio mv

Sirve para mover o renombrar archivos y directorios. Por ejemplo

```
$ mv a nueva
```

Esto lo que hace es renombrar el archivo, sin importar si hay ya anteriormente un archivo llamado “nueva”.

Ahora, vamos a ver como hacemos para que sea un poco más verborrágico. Esto se logra con la opcion “-v”

```
$ mv -v a nuevísimo
```

De esta manera vamos a ver qué es lo que hizo. Si queremos mover el archivo **nuevisímo** a la carpeta tmp:

```
$ mv -v nuevísimo /tmp/
```

Y lo mismo si queremos mover un directorio:

```
$ mv A B
```

Copiar archivos o directorios cp

Si hay un comando para mover o renombrar archivos. También va a existir un comando para copiar:

```
$ cp a copia_a
```

O sobrescribir un archivo

```
$ cp a b
```

Por otro lado, si quiero confirmación y verbosidad:

```
$ cp -iv a b
```

Y si quisiéramos copiar un directorio existe la opción "-r". Esta opción implica que lo va a hacer recursivo. Así que si yo quiero copiar una carpeta junto con todas las carpetas que hay dentro, ver lo que hace y que ante cualquier pregunta conteste que si haria

```
$ cp -rfv A NA
```

Si yo quisiera copiar la carpeta NA dentro de tmp, por ejemplo:

```
$ cp -rfv NA /tmp
```

Eliminar archivos y carpetas mediante el comando rm

Para borrar ejecutamos el comando rm. Este comando sigue la misma lógica de los anteriores. Si quiero borrar un archivo sin preguntar:

```
$ rm a
```

Si quisiera borrar un archivo y que me pregunte:

```
$ rm -i -v b
```

Pero no así con los directorios. Como en el copy, vamos a tener que poner también el -r si queremos borrar. Aprovechemos para que ponga un poco más de output:

```
$ rm -rfv NA
```

Leer archivos o cat,more, less

El primero es el cat que sirve para ver el contenido de los archivos de manera concatenada y sin paginar. Osea, todo seguido

```
$ cat /etc/passwd /etc/shadow
```

Un segundo acercamiento es el more. Lo que hace es paginar ante cada pantalla:

```
$ more /etc/passwd /etc/shadow
```

Por último, para tener un control total en el paginado al momento de mostrar los archivos utilizaremos **less**

```
$ less /var/log/dmesg
```

Enter o espacio bajo, flecha para arriba subo, y para los costados me muevo hacia los costados. Estos son los comandos para ver el contenido de un archivo.

Comandos head y tail para ver principio o fin del archivo

Ahora vamos a ver 2 comandos que me van a servir para ver la cola y el encabezado de un archivo. Primero arranquemos por el tail.

El tail me permite ver el final de un archivo. Por defecto va a mostrar las últimas 10 líneas:

```
tail /root/c
```

Si yo pongo:

```
tail -n 5 c
```

Voy a ver las últimas 5 líneas de c. Pero la verdadera utilidad la vamos a ver a continuación. Si yo pongo

```
tail -f c
```

Me va a mostrar las últimas 10 líneas; pero vean que se frenó. Este “frenado” lo que hace es que me muestra en tiempo real lo que se va agregando al archivo.

Para el comando **head** es como si se imaginaran la contraparte y sirve para ver los encabezados. Por defecto muestra las primeras 10 líneas. Ejemplo

```
$ head /var/log/syslog
```


Búsqueda de archivos dentro del filesystem o find

Busca archivos o carpetas en base a ciertos parámetros. Tiene muchísimo modificadores. Vamos a ver algunos:

```
$ find / -name B
```

Me trae todas las coincidencias de B. Ahora si yo quiero que sean solo los archivos. No directorios:

```
$ find /root -type f B
```

Filtrar la salida standard: grep

Lo que hace es mostrar la línea que coincide con ciertos caracteres. Por ejemplo:

```
$ grep root /etc/passwd
```

Estas son todas las líneas que matchean root en el **/etc/passwd**. Si hago un

```
grep -e root /var/log/auth.log
```

Me devuelve la cantidad de líneas que matchean con el patron que di. Por otro lado si yo no quiero que sea case sensitive agrego la opcion “-i”

```
$ grep RoOt /etc/passwd
```

```
$ grep -i RoOt /etc/passwd
```

```
$ grep -i -e roOT /etc/passwd
```

Otro comando es:

```
$ grep -H root /etc/passwd /etc/group /var/log/auth
```

Que me va a traer el archivo con las líneas correspondientes. Si uso el modificador -l solo me trae el nombre de los archivos en los que encontró coincidencia. Otro modificador interesante es el -n que, me trae el numero de linea:

```
$ grep -n root /var/log/auth.log
```

Y si quiero que me traiga las líneas donde NO esta root. Osea, lo opuesto que veniamos viendo antes, hacemos:

```
$ grep -v root /etc/passwd
```

Luego, si queremos que lo haga en forma recursiva indicándonos el archivo onde encontro la ocurrencia, usemos “-r”. Recursivo es cuando busca en el directorio que indicamos, y todos los superiores/inferiores. En este ejemplo voy a buscar el string **root** recursivamente en todos los archivos a partir del directorio **/var/log**.

```
$ grep -r root /var/log
```

Man o manual man

Por último, el man es el manual del comando. Si no recordamos parámetros o para que sirve ejecutando

```
$ man ls
```

Vamos a poder ver por ejemplo el manual de dicho comando. En este caso, el manual del comando ls.