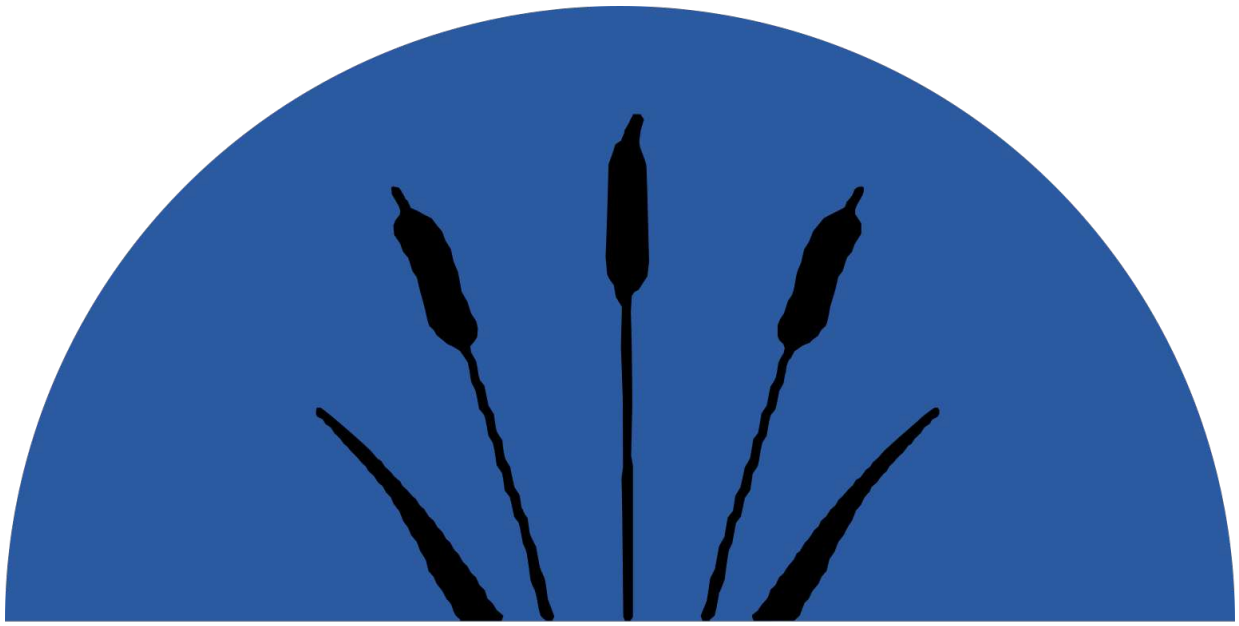

Junco TIC

GNU/Linux: Primeros pasos!

Guía de comandos básicos

Temario

Guía de comandos básicos

A continuación se planteará una breve guía práctica de comandos y ejemplos en terminal, con la intención de que puedas probarlos en tu sistema GNU/Linux y extiendas tus habilidades en la línea de órdenes!

Se presentan una serie de comandos y algunas opciones comunes, pero si quieres ir más allá y profundizar en otras opciones, no tienen mas que acudir al comando "**man**" para ello!

```
man pwd
man ls
man cd
```

Esta guía se irá **completando y actualizando**, por lo que te recomiendo que la revises, saques tus interrogantes con los comandos, consultes las dudas o problemas que te surjan, y por sobre todo, practiques el uso de comandos en tus tareas rutinarias en el sistema.

Por último, hay comandos que deben ser ejecutados con privilegios de root para que tengan efecto, tales como reboot o shutdown, por lo que en estos casos, si necesitas ejecutarlos, deberás utilizar "**su**" o "**sudo**" tal y como se vio en el curso.

pwd – Print working Directory

Imprime el directorio de trabajo actual.

Por defecto, este directorio es el directorio **home** del usuario, salvo que lo cambiemos con el comando "cd" (ver más adelante).

```
diego@cryptos:~$ pwd
/home/diego
```

cd – Change Directory

Cambia el directorio actual. El directorio destino puede ser cualquier directorio del sistema utilizando rutas absolutas o relativas.

```
diego@cryptos:~$ cd /tmp/
diego@cryptos:tmp$ pwd
/tmp
diego@cryptos:tmp$ cd /usr/share/doc/
diego@cryptos:doc$ pwd
/usr/share/doc
```

El modificador "-" permite volver al directorio inmediato anterior siempre que quisiéramos. Si ejecutamos este comando luego de los anteriores:

```
diego@cryptos:doc$ cd -
/tmp
diego@cryptos:tmp$ cd -
/usr/share/doc
```

ls – List files

Lista el contenido de los directorios. Sin argumentos, este comando lista el contenido del directorio de trabajo actual. Si no, lista el contenido del directorio especificado.

Algunos ejemplos:

```
diego@cryptos:doc$ ls /
bin   dev  home  lib64       media  opt   root  sbin  sys  usr
boot  etc  lib   lost+found  mnt    proc  run   srv   tmp  var

diego@cryptos:doc$ ls /home/
andy  diego  lost+found  user
```

Algunos modificadores interesantes de este comando son:

- **-l** para listar más información de las entradas (long format)
- **-a** para listar los archivos ocultos también (all files)
- **-d** para listar solamente el directorio del argumento (no su contenido). Muy útil si se combina con "-l".
- **-R** para listar recursivamente el contenido de un directorio y sus subdirectorios.
- **-S** ordena por tamaño.
- **-t** ordena por tiempo de modificación
- **-r** invierte el orden de listado.
- **-h** muestra los tamaños en formato legible (no solo bytes).

Un ejemplo interesante de estas combinaciones puede ser:

```
diego@cryptos:~$ ls -lhtr /home/
total 44K
```

```
drwx------   2 root  root   16K May 28  2014 lost+found
drwxr-xr-x   6 andy  andy  4.0K Jun  4  2015 user
drwxr-xr-x   9 andy  andy  4.0K Jul 29  2016 andy
drwxr-xr-x 142 diego diego  20K Nov 19 23:15 diego
```

du – Disk Usage

Permite ver el uso de disco por cada archivo en bytes. Podemos pasar un directorio por argumento para listar los tamaños de sus elementos interiores.

Opciones interesantes son "-**h**" para mostrar los tamaños en un formato "humano" más legible, y "**-s**" para *sumarizar* todos los tamaños de un directorio de argumento y mostrarnos el tamaño total del directorio, muy útil.

```
diego@cryptos:~$ du -sh /home/diego/Downloads/
99G  /home/diego/Downloads/
```

df – Disk Free

Permite visualizar el uso de cada uno de los sistemas de archivos montados. Aquí también es útil el modificador **"-h"** para mejorar la legibilidad de los tamaños.

```
diego@cryptos:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
dev             5.4G     0  5.4G   0% /dev
run             5.4G  736K  5.4G   1% /run
/dev/sda2        37G   34G  1.3G  97% /
tmpfs           5.4G  146M  5.2G   3% /dev/shm
tmpfs           5.4G     0  5.4G   0% /sys/fs/cgroup
tmpfs           5.4G  5.4M  5.4G   1% /tmp
/dev/sda5       878G  817G   43G  96% /home
tmpfs           1.1G   16K  1.1G   1% /run/user/1000
```

history – Historial de comandos

Permite ver los últimos comandos ejecutados por terminal, con un identificador numérico por cada uno.

```
diego@cryptos:~$ !517
history
    1  git pull
    2  ls thread/
    3  rm thread/a.out
    4  ls
    5  git push
    6  cd semaphore/
    7  ls
```

```
    8  gcc errorcontadorpractica.c
    9  gcc errorcontadorpractica.c -lrt
   10  gcc errorcontadorpractica.c -lpthread
   11  ./a.out
   12  vi errorcontadorpractica.c
   13  ls
   14  gcc posixsemaphore.c -lpthread
   15  ./a.out
   16  gcc posixsemaphorethread.c -lpthread
   17  ./a.out
   18  pwd
```

Luego, podemos hacer uso del comando **"!"** para correr alguno de los comandos listados, sin tener que escribirlos nuevamente. Por ejemplo, si en la salida anterior quisiéramos ejecutar el comando **pwd**, cuyo id es **18**, podemos hacer lo siguiente:

```
diego@cryptos:~$ !18
pwd
/home/diego
```

Por otro lado, para ejecutar el último comando de la lista, es decir, para repetir el último comando ejecutado, simplemente podemos hacer:

```
diego@cryptos:~$ !!
pwd
/home/diego
```

Esta opción es muy útil cuando trabajamos con privilegios de usuario común, y a veces debemos re-ejecutar un comando como root usando sudo:

```
diego@cryptos:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
diego@cryptos:~$ sudo !!
sudo cat /etc/shadow
```

(No, no voy a pegar aquí la salida del último comando :P).

mkdir – Make directory

Permite crear directorios en el sistema de archivos.

```
diego@cryptos:~$ mkdir /tmp/nuevo_directorio
diego@cryptos:~$ ls -ld /tmp/nuevo_directorio/
drwxr-xr-x 2 diego diego 40 Nov 20 19:12 /tmp/nuevo_directorio/
```

Un modificador interesante es **"-p"**, que permite crear un árbol jerárquico de

directorios inexistentes. Un ejemplo aclara las dudas:

```
diego@cryptos:~$ mkdir -p /tmp/nuevo_directorio/otro_mas/y_otro_nuevo
diego@cryptos:~$ tree /tmp/nuevo_directorio/
/tmp/nuevo_directorio/
└── otro_mas
    └── y_otro_nuevo
2 directories, 0 files
```

El uso del comando "**tree**" en este caso permite ver el arbol creado. Una salida similar podríamos haber obtenido con el modificador **"-R"** del comando **"ls"**:

```
diego@cryptos:~$ ls -R /tmp/nuevo_directorio/
/tmp/nuevo_directorio/:
otro_mas

/tmp/nuevo_directorio/otro_mas:
y_otro_nuevo

/tmp/nuevo_directorio/otro_mas/y_otro_nuevo:
```

rmdir – Remove directory / rm

Elimina directorios vacíos:

```
diego@cryptos:~$ rmdir /tmp/nuevo_directorio/otro_mas/y_otro_nuevo/
```

Si el directorio no está vacío el comando fallará:

```
rmdir: failed to remove '/tmp/nuevo_directorio/': Directory not empty
```

Para eliminar un directorio con contenido ver el comando "rm".

rm – Elimina archivos y directorios

Este comando permite eliminar archivos y directorios.

Algunos modificadores interesantes son:

- **-r** elimina un directorio en forma recursiva (directorios, subdirectorios y archivos).
- **-f** fuerza la eliminación de un directorio en el caso de que tenga contenido.

- **-d** elimina directorios vacíos.

shutdown

Permite apagar el sistema, o realizar alguna acción relacionada con la gestión de energía.

Permite especificar un argumento con la opción a ejecutar (apagar, reiniciar) y un temporizador.

Además permite enviar un mensaje a los usuarios que estén usando el sistema, ya sea en terminal nativa tty, o pseudoterminal remota.

La sintaxis es:

shutdown [opciones] tiempo [mensaje]

Algunas opciones interesantes son:

- **-r** reinicio (reboot)
- **-h** apagado (halt)
- **-k** envía un mensaje de apagado a los usuarios, pero no apaga.
- **-c** Cancela la orden de apagado temporizada.

El tiempo puede estar en alguno de estos formatos:

- **hh:mm** Realiza la acción en la hora y minuto especificados
- **minutos** Realiza la acción luego de X cantidad de minutos desde la ejecución.
- **now** Realiza la acción inmediatamente.

Veamos algunos ejemplos:

```
shutdown -h now      #apaga inmediatamente
shutdown -r now      #reinicia inmediatamente
shutdown -h 30       #apaga en media hora
shutdown -r 5 “Reinicio inminente”        #reinicia en 5 minutos
enviando un mensaje a los usuarios.
shutdown -h 02:00    #apaga el sistema a las 2 AM
```

halt / poweroff

Los comandos **halt** y **poweroff** también permiten apagar el sistema, aunque

no permiten especificar opciones adicionales, mensajes y temporizadores como el comando **shutdown**.

reboot

Reinicia el sistema inmediatamente.

cat

Permite ver el contenido de un archivo por terminal.

Si se especifican varios archivos, listará el contenido de todos ellos. Podemos ver el contenido de un archivo así:

```
cat archivo.txt
```

O listar el contenido de varios archivos así:

```
cat archivo1 archivo2 archivo3
```

Donde, por supuesto, podemos especificar las rutas absolutas de ubicación de cada archivo.

head

Este comando es similar a cat, pero lista por defecto las 10 primeras líneas de un archivo. Este comando es muy útil para "ver" cómo inicia un archivo particular, un código fuente, o documento de texto txt, etc.

Para especificar un número determinado de líneas desde el principio del archivo podemos usar el modificador "-n":

```
head /tmp/archivo.txt     #lista las primeras 10 líneas del archivo.
head /tmp/archivo.txt -n23 #lista las primeras 23 líneas del archivo.
```

Otro modificador interesante que puede llegar a ser útil es "-c" para listar los primeros N bytes o caracteres de un archivo de texto. Así:

```
head /tmp/archivo.txt -c 150    #lista los primeros 150 caracteres del
archivo.
```

tail

Idem al comando head, pero listando las ultimas líneas en vez de las primeras:

```
tail /tmp/archivo.txt     #lista las últimas 10 líneas del archivo.
tail /tmp/archivo.txt -n23 #lista las últimas 23 líneas del archivo.
```

Este comando también tiene el modificador **“-c”** para listar los últimos N bytes o caracteres del archivo.

Y por último, un modificador muy útil de tail es **“-f”**, o **“--follow”**, que permite listar las últimas líneas de un archivo, y quedar activo para listar las líneas que se vayan agregando dinámicamente. Esto es muy útil para listar archivos de log del sistema, y nos permite, a los sysadmines, ver en tiempo real los últimos mensajes que se van agregando:

```
tail -f /var/log/syslog
```

wc

Muestra el número de líneas, palabras o bytes de un archivo de texto

```
diego@cryptos:~$ wc -c /etc/fstab            # caracteres
543 /etc/fstab
diego@cryptos:~$ wc -w /etc/fstab            # palabras
44 /etc/fstab
diego@cryptos:~$ wc -l /etc/fstab            # lineas
13 /etc/fstab
```

ln

Permite numerar las líneas de un archivo (sin modificar el original). El archivo numerado se muestra por pantalla:

```
diego@cryptos:~$ cat /tmp/archivo1
Hola Mundo
que tal, Este
es un archivo de
texto plano
diego@cryptos:~$ nl /tmp/archivo1
     1    Hola Mundo
     2    que tal, Este
     3    es un archivo de
     4    texto plano
```

diff

Permite analizar las diferencias entre dos archivos de texto. Es un comando bastante utilizado en programación para comparar archivos de código fuente de aplicaciones.

```
diego@cryptos:~$ cat /tmp/archivo1
Hola Mundo
```

```
que tal, este
es un archivo de
texto plano
diego@cryptos:~$
diego@cryptos:~$ cat /tmp/archivo2
Hola Mundo
que tal este
es un archivo de
texto plano
diego@cryptos:~$
diego@cryptos:~$
diego@cryptos:~$ diff /tmp/archivo1 /tmp/archivo2
2c2
< que tal, este
---
> que tal este
```

find

Permite buscar archivos dentro de un sistema de archivos.

Find es una herramienta muy poderosa que da muchísimas facilidades a los administradores GNU/Linux.

La sintaxis básica del comando es:

```
find <ruta_donde_buscar> <patron_seleccion>
```

Entre los patrones de búsqueda tenemos:

- -**name** 'nombre'
  - Permite buscar por nombre de archivo. Si no se conoce el nombre completo, se pueden usar comodines, ej: 'archivo*.*'.
- -**iname** "nombre"
  - Similar a "-name", con la diferencia que no distingue mayúsculas de minúsculas.

    find -iname '*log_system*'
- -**size** +/- NN [c|k|w|b]
  - Permite buscar por tamaño. NN representa la cantidad, y las letras c, k, w y b representan bytes, kibibytes, words de dos bytes, o bloques de disco respectivamente.
  - Los signos + y – permiten especificar si el tamaño de los archivos debe ser mayor o menor que la cantidad establecida. Veamos algunos ejemplos:
    find /home -size +1024k #archivos de mas de 1MiB
    find /home -size +100b -size -1k #archivos de mas de 100b y

menos de un kibibyte.

- -type <tipo>
  - Permite buscar por tipo de archivo, donde los tipos válidos son:
    - **d** directorio
    - **f** archivos regulares / files
    - **l** symlink o enlaces simbólicos
    - **b** dispositivos de bloque
    - **c** dispositivos de carácter
    - **p** pipes
    - **s** sockets

find permite especificar muchas opciones más, entre ellas, -**perm** para buscar por permisos de archivos, -**exec** para ejecutar un comando con cada resultado de búsqueda, -**nouser** y -**nogroup** para buscar por id de usuario y grupo, etc. En ***futuras actualizaciones*** agregaremos más ejemplos del uso de find!

whereis

Muestra información de un comando o binario. Muestra el archivo binario que ejecuta el comando, archivos fuente si los hay, archivos de **man-pages** si están cargados, etc.

```
diego@cryptos:~$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1p.gz
/usr/share/man/man1/ls.1.gz
```

which

Similar a **whereis**, pero solamente muestra el directorio donde se encuentra el binario que invoca.

```
diego@cryptos:~$ which ls
/usr/bin/ls
```

grep

Busca patrones de texto dentro de archivos. Es una herramienta súper extensa y útil para un sysadmin!

Por defecto devuelve todas las líneas que contienen el patrón de búsqueda, aunque pueden especificarse algunas opciones interesantes:

- **-c** solo devuelve la cantidad de líneas con el patrón

- **-i** ignora mayúsculas y minúsculas
- **-H** Imprime el nombre del archivo además de la línea. Esta es la opción por defecto cuando se busca en más de un archivo.
- **-l** solamente muestra el nombre del archivo donde se encontró el patrón.
- **-r** realiza búsquedas recursivas en más de un directorio
- **-n** imprime el número de la línea donde se encontró el patrón
- **-o** muestra solamente el patrón que encontró, no la línea entera.

Veamos algunos ejemplos:

```
diego@cryptos:~$ cat /tmp/archivo1
Hola Mundo
que tal, este
es un archivo de
texto plano
diego@cryptos:~$ grep 'es' /tmp/archivo1
es un archivo de
diego@cryptos:~$ grep 'es' /tmp/archivo1 -i
que tal, Este
es un archivo de
diego@cryptos:~$ grep 'es' /tmp/archivo1 -c
1
diego@cryptos:~$ grep 'es' /tmp/archivo1 -c -i
2
diego@cryptos:~$ grep 'es' /tmp/archivo1 -i -l
/tmp/archivo1
diego@cryptos:~$ grep 'es' /tmp/archivo1 -i
que tal, Este
es un archivo de
diego@cryptos:~$ grep 'es' /tmp/archivo1 -i  -n
2:que tal, Este
3:es un archivo de
diego@cryptos:~$ grep 'es' /tmp/archivo1 -i  -n -o es
/tmp/archivo1:2:Es
/tmp/archivo1:3:es
```

NOTA:

Esta guía representa una seria de comandos útiles que todo operador y administrador GNU/Linux debería saber para mejorar sus habilidades en el uso del sistema.

Con el tiempo se irá actualizando esta guía así como también el contenido del curso, por lo que te recomiendo que practiques estos comandos y evacues tus dudas!

Cualquier inconveniente que te surja no tienes mas que escribirme! Y por supuesto, si detectas algún error o tienes alguna sugerencia de comando a agregar a esta guía, también escríbeme así lo tengo en consideración para las siguientes actualizaciones! GRACIAS!

Comunidad JuncoTIC

Recuerda que puedes sumarte a la **Comunidad JuncoTIC** siguiéndonos en nuestras redes sociales!

Facebook **Twitter** **Linkedin** **Google+** **Youtube** **Telegram**

Puedes aprovechar todo el contenido que diariamente estamos publicando y actualizando en nuestro blog!

https://juncotic.com/blog/

Además, te invitamos a **suscribirte** a nuestra lista de distribución para que recibas todas las semanas novedades de contenidos publicados en el blog, y últimos lanzamientos.

Por cualquier duda, sugerencia de información sobre cursos y temarios, o información sobre nuestros servicios, esperamos tu consulta en:

https://juncotic.com/contacto

O directamente por correo electrónico a info@juncotic.com