

Comenzando con el Intérprete de comandos bash

Índice de contenido

INTRODUCCIÓN.....	3
CONSIDERACIONES INICIALES.....	3
COMANDOS USUALES.....	4
INCLUÍDOS EN BASH.....	5
MANEJO DE ARCHIVOS.....	5
USUALES.....	6
ADMINISTRACIÓN.....	8
PERMISOS.....	8
FILTROS.....	8
SOBRE BASH.....	9
COMODINES	9
COMPLETAMIENTO AUTOMATICO.....	10
ENTRADA/SALIDA(I/O).....	10
/dev/null.....	11
PROCESOS Y TAREAS.....	11
ALIAS.....	11
REFERENCIAS.....	12

INTRODUCCIÓN

Durante mucho tiempo he leído en foros, mensajes de listas de correo y escuchado comentarios afirmando que el intérprete de comandos o consola es muy importante en el uso un sistema operativo(en adelante SO) GNU/Linux ¹. Al inicio(cuando estaba probando el SO) me aventuré a leer un poco, pero luego lo dejé debido a que existen una gran cantidad de comandos y lo que me interesó en un primer momento fue realizar labores tal como lo hacía con otro SO que usaba.

Sin embargo varias veces he llegado otra vez a la consola tratando de resolver algún problema debido a que las soluciones que encontraba eran para otras distribuciones; y tomando en cuenta que cada una suele tener formas de administración gráficas diferentes el intérprete de comandos se convierte en una especie de idioma universal.

Otra de las razones que me lleva a la línea de comandos de vez en cuando es la búsqueda de un espacio de trabajo más rápido mientras la PC se encuentra realizando mucho trabajo, ya que el entorno gráfico puede llegar a desesperar un poco si el tiempo de respuesta es muy largo; y no siempre se tiene una computadora con mucha potencia. Otro de los motivos que me ha hecho utilizarla es la necesidad de "matar" un programa que no responde rápidamente.

El siguiente artículo no pretende ser una guía completa de comandos disponibles ni abarcar todas las funcionalidades de cada uno de éstos, sino sólo mostrar algunos comandos básicos que podrían introducir a niveles mayores de manejo de la consola.

CONSIDERACIONES INICIALES

Si bien existen varios intérpretes de comandos, el más difundido en SO's GNU/Linux es el bash².

Para ingresar a trabajar en la línea de comandos no es necesario estar en el modo texto. Casi todas las distribuciones³ cuentan con una consola en modo gráfico que suele accederse desde el inicio rápido, el menú de herramientas o el lanzador de aplicaciones(usualmente presionando ALT+F2). En el caso del escritorio GNOME se cuenta con gnome-terminal y en el caso de KDE, con konsole; por mencionar dos ejemplos del modo gráfico.

La otra opción es ingresar en modo texto, ésto se logra con la combinación de teclas CTRL+ALTFX(si se ha ingresado en modo gráfico); donde FX es una tecla de función de la F1 a la F7 (normalmente, pero puede que haya un número diferente de consolas disponibles). Luego de ésto, necesitaremos ingresar como usuarios poniendo el login y la contraseña habituales.

En general en la consola se puede encontrar comandos que siguen el siguiente esquema:

```
NombreComando [-opciones [argumento]]
```

Notas: Los corchetes indican que lo que hay dentro es prescindible. Cabe mencionar que los comandos difieren si están escritos con mayúsculas o minúsculas(case sensitive).

1 <http://es.wikipedia.org/wiki/GNU/Linux>

2 <http://es.wikipedia.org/wiki/Bash>

3 http://es.wikipedia.org/wiki/Distribuci%C3%B3n_Linux

A continuación encontrarán algunas formas de llegar a una ayuda desde consola, y justamente es a través de comandos.

whatis

Nos muestra en una línea la funcionalidad de un comando. Ayudará mucho si por cada comando nuevo se ejecuta `whatis`, pasándole como argumento el nombre del comando del cual se quiere obtener información.

```
home@home-desktop:~$ whatis whatis
whatis (1)          - display manual page descriptions
```

whereis

Mostrará el(los) directorio(s) donde se encuentra almacenado el comando, el código(si estuviera) y el manual.

```
home@home-desktop:~$ whereis whatis
whatis: /usr/bin/whatis /usr/bin/X11/whatis
/usr/share/man/man1/whatis.1.gz
```

man

Permite acceder al manual del comando dado. Especifica detalladamente todas las formas de uso con las respectivas opciones.

```
man -k texto_a_buscar
```

Busca "texto_a_buscar" en alguna cabecera de manual, devolviendo los comandos que lo contienen. También pueden utilizar `apropos` en vez de `man -k`.

COMANDOS USUALES

Existen muchos manuales y tutoriales donde se pueden encontrar conjuntos de comandos "básicos" hechos por diferentes autores que mas o menos coinciden.

Dejo aquí algunos:

http://es.tldp.org/Tutoriales/CURSOLINUX/curso_linux/node64.html

<http://es.tldp.org/Manuales-LuCAS/LIPP/lipp-1.1-html-2/lipp3.htm#3.4>

También pueden encontrar inmensas listas de comandos como estas dos:

<http://www.ss64.com/bash/index.html>

http://agora.pucp.edu.pe/ftp/pub/linux/manuales/otros/linux_command_ref.pdf

Sin embargo me animo a presentarles una propia micro-recopilación.

INCLUÍDOS EN BASH

echo

Muestra texto.

```
home@home-desktop:~/dir1$ echo hola
hola
```

cd

Cambia de directorio.

```
home@home-desktop:~$ cd dir1/
(nos lleva al directorio dir1 de nuestra carpeta de usuario)
```

history

Muestra la lista de los HISTSIZE(tamaño del historial) comandos digitados. Podemos usar !N para ejecutar una línea anterior otra vez, reemplazando N por el número de la entrada que aparece luego de ejecutar el comando.

pwd

Imprime la dirección del directorio actual.

```
home@home-desktop:~$ pwd
/home/home
```

MANEJO DE ARCHIVOS

mkdir

Crea directorios.

```
home@home-desktop:~$ mkdir dir1
home@home-desktop:~$ mkdir subdir{1..4}/archivo{A..E}.dat
```

cp

Copia ficheros,

```
home@home-desktop:~$ cp dir1/subdir1/* dir2/
```

mv

Mueve ficheros.

```
home@home-desktop:~/dir2$ mv archivoA.dat archivoX.dat
(en este caso renombra)
```

rm

Elimina ficheros.

```
home@home-desktop:~/dir2$ rm archivo*
```

rmdir

Elimina directorios vacíos.

file

Muestra el tipo de archivo.

```
home@home-desktop:~$ file mailbox
mailbox: ASCII mail text
```

USUALES

ls

Lista contenidos de un directorio

```
home@home-desktop:~/dir1$ ls
archivo1.dat  archivo3.dat  archivo5.dat  archivo7.dat  subdir2
subdir4
archivo2.dat  archivo4.dat  archivo6.dat  subdir1       subdir3
```

clear

Limpia la pantalla actual.

find

Encuentra archivos.

```
home@home-desktop:~/dir1$ find -name 'arch*A*'
./subdir1/archivoA.dat
./subdir2/archivoA.dat
./subdir3/archivoA.dat
./subdir4/archivoA.dat
```

more y less

Permiten desplazarse a través de páginas de información. Usualmente accedidas por tuberías.

```
home@home-desktop:~/dir1$ ls /bin/ | less
```

cat

Concatena e imprime en la salida estándar. La combinación CTRL+D representa el fin de archivo(EOF).

```
home@home-desktop:~/dir1$ cat archivo1.dat
uno
dos
tres
cuatro
cinco
```

nano, pico, vi

Editores de texto.

touch

Cambia las fechas de archivos; y si no existen los crea.

date

Permite mostrar la fecha actual en formatos personalizados.

```
home@home-desktop:~$ date +'Fecha ejemplo: '%A$' '%d$' de '%B$'
del '%Y$'\nSon: '%T$' hs'
Fecha ejemplo: martes 15 de agosto del 2006
Son: 09:03:29 hs
```

ln -s

Crea enlaces.

```
home@home-desktop:~$ ln -s marvil/tuxpuc/ tuxpuc
```

tar

Herramienta para conglomerar ficheros.

```
home@home-desktop:~/dir1$ tar -cvf archivo.tar archivo*
(junta todos los ficheros que comienzan por archivo y los añade a archivo.tar)
```

gzip

Administra la compresión de ficheros.

```
home@home-desktop:~/dir1$ gzip archivo.tar
(comprime)
```

```
home@home-desktop:~/dir1$ gzip -d archivo.tar.gz
(descomprime)
```

Para juntar en un solo archivo y comprimir al mismo tiempo podemos usar:

```
home@home-desktop:~/dir1$ tar -czvf archivo.tar.gz archivo*
archivo1.dat
archivo2.dat
archivo3.dat
archivo4.dat
```

Y para descomprimir y desempaquetar al mismo tiempo podemos usar:

```
home@home-desktop:~/dir1$ tar -xzvf archivo.tar.gz
archivo1.dat
archivo2.dat
archivo3.dat
archivo4.dat
```

ADMINISTRACIÓN

sudo

Permite ejecutar un comando desde como otro usuario.

```
home@home-desktop:~/dir1$ sudo -u root nano
/etc/apt/sources.list
```

su

Permite convertirse en el súper-usuario.

PERMISOS

chmod

Cambia los permisos de un fichero.

```
home@home-desktop:~/dir1$ chmod u+x archivo.tar.gz
```

chown

Cambia el propietario de un fichero.

```
home@home-desktop:~/dir1$ sudo chown -R root subdir1/
```

FILTROS

sort

Ordena las líneas de la entrada que recibe.

```
home@home-desktop:~/dir1$ cat archivo1.dat | sort
cinco
cuatro
dos
tres
uno
```

paste

Une líneas de diferentes archivos.

```
home@home-desktop:~/dir1$ paste archivo1.dat archivo2.dat
uno      seis
dos      siete
tres     ocho
cuatro   nueve
cinco    diez
```

tac

Concatena e imprime archivos invertidos.

```
home@home-desktop:~/dir1$ tac archivo1.dat
cinco
cuatro
tres
dos
uno
```

tr

Traduce o borra caracteres

```
home@home-desktop:~/dir1$ tr nh ñ
anho canha
año caña
```

WC

Cuenta bytes, palabras y líneas

grep

Muestra las coincidencias con la entrada que se le brinde.

```
home@home-desktop:~/dir1$ dpkg -l | grep klipper
ii  klipper                                3.5.2-0ubuntu27
    clipboard utility for KDE
```

SOBRE BASH

En el intérprete de comandos existen algunas consideraciones que pueden sernos muy útiles. Aquí trato de recopilar algunas de ellas.

COMODINES

*

* -> reemplazar por cualquier caracter o conjunto de caracteres

a* -> reemplazar por todo lo que se inicie con a

*a -> reemplazar por todo lo que termine por a

a*b -> reemplazar por todo lo que comience por a y termine por b

a -> reemplazar por todo lo que contenga una a en la palabra

?

? -> reemplazar por un caracter cualquiera

?a -> reemplazar por lo que tenga 2 caracteres y termine en a

a? -> reemplazar por lo que tenga 2 caracteres y comience en a

[]

[xyz] -> uno de los caracteres es reemplazado a la vez en esa posición

COMPLETAMIENTO AUTOMATICO

(TAB)

Para establecer el nombre de un archivo o directorio dentro de un comando se puede usar el completamiento automático, tan sólo presionar la tecla TAB en algún momento. si hay una sola opción ésta se muestra, de otra manera pulsando por segunda vez TAB se obtiene la lista de las posibilidades y la línea de comando que se hubo escrito.

ENTRADA/SALIDA(I/O)

La línea de comandos utiliza la entrada y salidas estándar. Ésta es una de las razones en la facilidad de su uso.

1. Redireccionamiento

Los caracteres '<' y '>'; permiten el redireccionamiento de la entrada y salida respectivamente.

Por ejemplo podríamos copiar archivos sin sólo con el comando cat:

```
home@home-desktop:~$ cat arch1 > arch2
```

Copiaría el contenido de arch1 en arch2.

2. Tuberías

El caracter '|'; permite transmitir información de un comando a otro a continuación del término de la ejecución del otro.

Por ejemplo para mostrar ordenado alfabéticamente por líneas un archivo de texto usaríamos:

```
home@home-desktop:~$ cat arch1 | sort
```

De esta manera todo lo que debería mostrar el comando cat en la salida estándar es pasado al siguiente comando, sort, que está ordenándola.

/dev/null

Es un archivo que emula un dispositivo como un "agujero negro".

PROCESOS Y TAREAS⁴

Para mantener en ejecución una tarea en segundo plano añadimos & al final. Luego de iniciar la tarea nos devuelve el número de tarea y su correspondiente PID.

```
home@home-desktop:~$ yes > /dev/null &
[1] 8690
```

Algunas veces queremos parar el programa que se está ejecutando en primer plano por alguna razón. Tenemos dos opciones:

- Detenerlo con CTRL+C
- Pausarlo y pasarlo a segundo plano con CTRL+Z

```
home@home-desktop:~$ yes > /dev/null
```

Para ver las tareas actuales usamos jobs.

```
home@home-desktop:~$ jobs
[1]+  Stopped                  yes >/dev/null
```

Para regresar a la ejecución en primer plano de una tarea pausada usamos fg.

```
home@home-desktop:~$ fg %1
yes >/dev/null
(En este caso nos lleva a la tarea 1)
```

Otra manera de visualizar lo que se encuentra en ejecución es con el comando top, que suele ser útil para identificar qué está consumiendo más RAM o CPU y desde allí, si es que es necesario, terminarlo.

ALIAS

Bash nos permite crear alias o nombres auxiliares para instrucciones en la línea de comandos de manera sencilla:

```
alias [-p] [nombre[=valor] ... ]
```

Por ejemplo si deseáramos crear un alias para alias 'ls -l --color=auto' con sólo digitar 'l' lo haríamos así:

```
alias l='ls -l --color=auto'
```

Sin embargo esto sólo nos serviría para la sesión en la que nos encontramos. Para poder usar siempre alias debemos de editar el archivo .bashrc de nuestra carpeta de usuario añadiendo tantas líneas como alias deseemos o incluirlos todos en un archivo y llamarlos desde .bashrc añadiendo la línea ". nombre_archivo_con_crea_alias"(con lo que estaremos ejecutando las líneas que se encuentre en dicho archivo como si estuviésemos ingresándolas mediante la consola).

⁴ http://structio.sourceforge.net/guias/AA_Linux_colegio/procesos-y-tareas.html

REFERENCIAS

Para la elaboración del artículo se utilizaron varias fuentes que menciono a continuación.

- (Manual oficial de bash)
 - <http://www.gnu.org/software/bash/manual/bash.html>
- (The Linux users' guide)
 - <http://agora.pucp.edu.pe/ftp/pub/linux/manuales/otros/user-guide.pdf>
- (hora 14 y 1 anterior)
 - <http://agora.pucp.edu.pe/ftp/pub/linux/manuales/otros/UNIX.PDF>
- (43-102) Sams' teach yourself Linux in 24 hours
 - http://agora.pucp.edu.pe/ftp/pub/linux/manuales/otros/linux_24h.pdf
- (45-63) Aprenda Linux como si estuviera en primero
 - <http://mat21.etsii.upm.es/ayudainf/aprendainf/Linux/Linux.pdf>
- (60-73) Novato a novato
 - <http://es.tldp.org/Tutoriales/NOVATO/novato-a-novato/>
- (3.3 procesos y tareas)
 - http://structio.sourceforge.net/guias/AA_Linux_colegio/
- (lista inmensa de comandos por categorías)
 - http://agora.pucp.edu.pe/ftp/pub/linux/manuales/otros/linux_command_ref.pdf
- tldp
 - http://es.tldp.org/Tutoriales/CURSOLINUX/curso_linux/node16.html
 - <http://www.imago.com.uy/howto/linuxadmin.pdf>
- comandos básicos
 - http://es.tldp.org/Tutoriales/CURSOLINUX/curso_linux/node64.html