

Una sesión en Linux

Contenido

El emulador de terminal.....	1
Cómo escribir comandos.....	2
Sintaxis de los comandos.....	3
Páginas man.....	4
Directorios.....	5
Archivos.....	5
Listado de archivos.....	5
Contenido de un archivo.....	6
Crear y borrar archivos.....	7
Usuarios y grupos.....	8
Cambio de contraseña.....	9
Caracteres especiales.....	9
Referencias, lecturas complementarias.....	9

Este capítulo describe una primera sesión en Linux. Su objetivo es mostrar el uso de la interfaz de usuario de línea de comandos, interactuando con el shell `bash` (Bourne Again Shell). Se describen someramente algunos comandos comunes. La lectura de este capítulo debiera acompañarse de la experimentación en una terminal Linux, repitiendo los comandos aquí mostrados, observando su efecto.

El emulador de terminal

La interfaz de línea de comandos se presenta normalmente como un *emulador de terminal*, también llamada *terminal de caracteres* o solo *terminal*. Este programa emula un dispositivo con video y teclado donde el usuario puede indicar tareas a un computador escribiendo líneas sucesivas de texto llamadas *líneas de comandos*, recibiendo los resultados a continuación en líneas sucesivas de texto.

En un sistema Linux con interfaz gráfica, es posible invocar un terminal de alguna de estas formas:

- Desde el menú: invocar el menú, con el ícono de menú o con la tecla de MS Windows; buscar escribiendo la palabra "terminal", y aparecen una o varias opciones.
- Desde el menú, usualmente en la categoría Accesorios, aparecen una o más aplicaciones de terminal.
- Con las teclas Ctrl-Alt-T en varios sistemas de ventanas de Linux se despliega una terminal.

Es posible también usar las *consolas virtuales* o *terminales virtuales* del computador, interfaces de línea de comando que no requieren del ambiente gráfico. Las consolas virtuales numeradas de 1

a 6 están disponibles para trabajar en línea de comandos; la consola virtual 7 u 8 es donde suele encontrarse la interfaz gráfica. En las máquinas sin ambiente gráfico instalado, es necesario usar estas consolas virtuales.

- para usar una consola virtual, digitar las teclas Ctrl-Alt-1 a Ctrl-Alt-6.
- para volver al ambiente gráfico, digitar las teclas Ctrl-Alt-7 o en su defecto Ctrl-Alt-8.

Para cerrar una consola o terminal, digitar el comando `exit` o las teclas Ctrl-D.

El ingreso a una terminal virtual pedirá un nombre de usuario y contraseña, tal como lo hace el ingreso a la máquina en el ambiente gráfico.

Al iniciar una terminal, suele verse un mensaje de este tipo:

```
victor@uma ~ $
```

La palabra `victor` es el nombre de usuario, la arroba `@` indica en qué máquina, `uma` es el nombre de la máquina, `~` indica el directorio actual (en este caso el directorio propio del usuario), `$` es el indicador de comando, el usuario puede escribir un comando a continuación. El símbolo `$` indica que el usuario no es privilegiado; un superusuario vería el símbolo `#` en lugar de `$`.

Algunas advertencias:

- los sistemas Linux, al igual que todos sus antecesores Unix, distinguen entre letras mayúsculas y minúsculas. A estos efectos, los caracteres `a` y `A` son dos caracteres distintos, aunque sean la misma letra.
- los sistemas Unix/Linux, privilegian el uso de minúsculas: la inmensa mayoría de los comandos se escriben con minúscula, y es habitual escribir también con minúscula los identificadores o nombre de los usuarios en el sistema. Así, los nombres de usuario `AGarcia`, `agarcia`, `AGARCIA` son todos nombres distintos; el de preferencia sería `agarcia`.
- los nombres de usuario suelen escribirse con letras minúsculas, números y algunos caracteres como `-`, `_` o `.` (punto). Las contraseñas, para ser seguras, deben tener al menos 8 caracteres, y combinar letras minúsculas, al menos una letra mayúscula en posición no inicial, números y caracteres especiales; las cuatro categorías deben estar presentes en una buena contraseña.
- el usuario debe escribir cada comando con toda exactitud, letras, símbolos, espacios, todo importa. Luego de digitar el comando, se lo envía al sistema digitando la letra Enter.
- la ejecución de un comando puede interrumpirse digitando Ctrl-C. La sesión puede terminarse digitando Ctrl-D.

Cómo escribir comandos

Los siguientes comandos dan información sobre usuarios y sobre el sistema. Abrir una terminal y digitarlos exactamente como aparecen; examinar los resultados. Se indica brevemente el objetivo de cada uno, sin profundizar en su aplicación; se proponen solo como ejemplos del formato que adoptan los comandos.

Comando	Efecto
<code>echo Hola Mundo</code>	presenta en pantalla el texto indicado
<code>whoami</code>	muestra el nombre de usuario del operador
<code>who</code>	muestra los usuarios activos en el sistema, sus terminales, fecha, tiempo de actividad, otros datos.
<code>id</code>	datos del usuario operador: identificadores de usuario y grupos a los que pertenece
<code>id agarcia</code>	datos del usuario <code>agarcia</code>
<code>id --help</code>	ayuda del comando <code>id</code>
<code>echo \$LOGNAME</code>	nombre del usuario, interroga variable de ambiente
<code>echo \$HOMEDIR</code>	directorio propio del usuario, interroga variable de ambiente
<code>hostname</code>	nombre de la máquina.
<code>uname -a</code>	nombre de máquina, sistema operativo, versión, fechas, otros datos
<code>pwd</code>	el directorio donde se está operando.
<code>date</code>	fecha, hora, zona horaria.
<code>ls -l /tmp</code>	contenido detallado del directorio <code>/tmp</code> .
<code>ps</code>	procesos del usuario en ejecución; aparecen <code>bash</code> y el propio <code>ps</code> .
<code>ps aux</code>	procesos en ejecución en el sistema.
<code>ps aux less</code>	lo mismo, por páginas; con la barra espaciadora avanza una página, con la letra <code>q</code> termina.
<code>ps aux wc -l</code>	cantidad de procesos en ejecución en el sistema.
<code>man whoami</code>	explicación del comando <code>whoami</code> ; <code>man</code> viene de manual. Para salir, digitar la tecla <code>q</code> .
<code>man man</code>	explicación del comando de ayuda, <code>man</code> .
<code>cat /etc/hostname</code>	muestra contenido del archivo <code>hostname</code> en el directorio <code>/etc</code> .
<code>exit</code>	terminar la sesión, cierra el terminal.

Sintaxis de los comandos

Los comandos empiezan con el nombre del comando, la primera palabra. Pueden adjuntarse opciones como `aux` en `ps aux`, o `-a` en `uname -a`. Algunos comandos reciben, luego de las opciones, nombres de archivos, como `/etc/hostname` en `cat /etc/hostname`, o de usuarios, como en `id agarcia`. Según el comando, también pueden recibir nombres de directorios, dispositivos, identificadores de procesos, u otros. La barra vertical permite concatenar comandos, pasando la salida de un comando hacia el siguiente, como en `ps aux | wc -l`: la salida del primer comando es la lista de procesos en ejecución, el segundo comando cuenta la cantidad de líneas.

En informática, se denomina *sintaxis* al conjunto de reglas que definen las secuencias correctas de símbolos en un lenguaje de programación. La sintaxis de los comandos se puede ver en las "páginas man".

Páginas man

El comando `man` permite acceder a las "páginas man", documentación de referencia para comandos del sistema operativo, bibliotecas, llamadas al sistema, estándares, entre otros. La página man de cada comando describe la sintaxis de ese comando: nombre, opciones, nombres de directorios, archivos, dispositivos, usuarios u otros según el comando.

Las páginas man suelen estar en inglés; si bien hay traducciones, no suelen estar actualizadas. El comando

```
man cp
```

produce la siguiente salida:

```
CP (1)                                User Commands                                CP (1)

NAME
    cp - copy files and directories

SYNOPSIS
    cp [OPTION]... [-T] SOURCE DEST
    cp [OPTION]... SOURCE... DIRECTORY
    cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
    Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --archive
        same as -dR --preserve=all

    --attributes-only
        don't copy the file data, just the attributes

    --backup[=CONTROL]
        make a backup of each existing destination file

(siguen líneas)
```

En este extracto de la página man del comando `cp` :

<code>cp</code>	nombre del comando, abreviatura de "copy".
<code>[OPTION]</code>	indica una opción de las mostradas más abajo (-a, --archive, etc).
<code>...</code>	indica repetición opcional de lo anterior.
<code>SOURCE</code>	el archivo o directorio origen, determinado por el operador.
<code>DEST</code>	el archivo o directorio de destino, determinado por el operador.
<code>DIRECTORY</code>	un directorio.
<code>-a</code>	opción de archivo, versión corta.
<code>--archive</code>	opción de archivo, versión larga, más legible.
<code>--backup[=CONTROL]</code>	opción larga con valor opcional de control.

Las páginas man son la referencia obligada para el uso de cada comando; allí se describe su funcionamiento y todas las opciones. Las páginas man son un material de referencia, no necesariamente de estudio; su contenido técnico y totalmente abarcativo puede hacerlas difíciles de interpretar. No obstante, se recomienda consultar las páginas man ante cada nuevo comando, en particular su objetivo, la sintaxis, y las opciones correspondientes a lo que se quiera hacer.

Directorios

Un *directorio* es un contenedor para otros archivos o directorios; cada directorio se identifica con un nombre; en MS Windows al directorio se le llama *carpeta*. Como un directorio puede contener otros directorios, se genera un árbol de directorios; la raíz (el origen) de este árbol se designa con `/`; `/home` es un directorio de primer nivel, igual que `/tmp`; `/home/agarcia` es el directorio del usuario `agarcia`, bajo el directorio `home`, a su vez subdirectorio del directorio raíz designado con `/`. En Unix/Linux no se visualizan los dispositivos físicos, no hay C: ni A: como en MS Windows; todos los directorios entroncan en un mismo árbol de directorios, aunque físicamente se encuentren en dispositivos diferentes.

Cada usuario tiene un directorio propio para sus datos, llamado a veces "directorio home". Los directorios de los usuarios se encuentran usualmente bajo el directorio `/home`, como el `/home/agarcia` anterior. Cuando el usuario ingresa al sistema ya está ubicado en su directorio propio. El comando

```
pwd
```

muestra el directorio actual.

```
cd /home
```

cambia hacia el directorio `/home`, lo que puede verificarse con el comando `pwd`.

```
cd
```

sin parámetros devuelve al usuario a su directorio propio, desde cualquier lugar donde esté. Este comando es útil cuando se han hecho varios cambios de directorio y se quiere retornar a una situación conocida, ubicándose en el directorio propio.

Archivos

Listado de archivos

Ensayemos el comando para listar archivos y directorios:

```
ls
```

lista archivos del directorio actual, y los subdirectorios si los hay en el directorio actual.

```
ls -l /bin
```

lista archivos en el directorio `/bin`; aquí se encuentran los archivos de comandos ejecutables del sistema. No cambia de directorio; el directorio actual sigue siendo el mismo.

```
ls -l
```

lista archivos del directorio actual en formato largo, dando detalles; la opción se lee "menos ele".

La salida obtenida consta de renglones parecidos a

```
-rw-rw-rw- 1 esteban users 138 Apr 5 19:34 leame
```

cuyos elementos se interpretan así:

- indica el tipo de archivo de que se trata, con esta convención:

- archivo común,

d directorio,

l enlace o referencia a otro archivo.

`rw-rw-rw` son los permisos del archivo, con esta convención:

r (read) permiso para leer el archivo

w (write) permiso para modificar o eliminar el archivo

x (execute) si se trata de un archivo, permiso para ejecutarlo como programa; si se trata

de un directorio, permiso para ingresar en él y recorrerlo.

Los tres grupos de 3 caracteres indican permisos para el dueño del archivo (esteban), su grupo

(users) y el resto del mundo.

1 cantidad de enlaces, referencias a este archivo desde otros archivos ubicados en diferentes lugares.

esteban nombre del usuario dueño del archivo.

users nombre del grupo al que pertenece el archivo.

138 tamaño en bytes del archivo. Si se trata de un directorio, este número es de control del sistema, sin interpretación inmediata para el usuario.

Apr 5 19:34 fecha y hora de última modificación. Si no aparece el año, se asume el año corriente.

leame nombre del archivo. Notar que el nombre del archivo está siempre al final.

```
ls -a
```

muestra también archivos ocultos, normalmente no visibles en el listado. Los archivos cuyo nombre empieza con un punto son ocultos, en este sentido. Las entradas `.` y `..` representan el directorio actual y el directorio padre, respectivamente.

```
ls -la
```

combina las dos opciones, muestra formato largo y archivos ocultos.

```
ls -la /var
```

listado de archivos visibles y ocultos en formato largo del directorio `/var`.

Contenido de un archivo

Es posible guardar la salida de un comando en un archivo. Por ejemplo,

```
man man > man.txt
```

guarda el contenido de la página `man` del comando `man` en el archivo `man.txt`. Es un archivo de texto, pero la extensión `.txt` no tiene significado particular, es solo una convención para recordar el tipo de contenido. En Unix/Linux los nombres de archivo no tienen ninguna limitación, ni tienen por qué tener extensión.

```
cat man.txt
```

muestra el contenido del archivo.

```
less man.txt
```

muestra el mismo contenido pero página a página. Las teclas `Avpág` y `Repág` (`PgDn` y `PgUp`) avanzan y retroceden una página; la tecla `q` termina el despliegue. También es posible avanzar una página con la barra espaciadora, y retroceder con `Ctrl-B`.

```
cat man.txt | less
```

hace lo mismo concatenando dos comandos: la salida de `cat` es enviada a `less`, que presenta la información página por página.

```
head man.txt
```

muestra las primeras 10 líneas de un archivo.

```
tail -5 man.txt
```

muestra las 5 líneas finales de un archivo; `head` también soporta la opción de indicar cantidad de líneas a mostrar.

```
ls -l /etc | less
```

muestra el extenso contenido del directorio `/etc` paginando la salida.

Crear y borrar archivos

```
touch nota.vacia
```

crea el archivo `nota.vacia`, sin contenido alguno.

```
ls -l nota.vacia
```

muestra datos del archivo creado, verifica que está allí.

```
cat nota.vacia
```

no muestra nada, el archivo está vacío.

```
touch .archivo_oculto
```

crea un archivo vacío y oculto. Los archivos cuyo nombre comienza por un punto no son visibles a no ser que se indique, por ejemplo con la opción `-a` en `ls`:

```
ls
```

no muestra el archivo oculto creado, pero

```
ls -a
```

sí lo muestra. Las opciones se pueden combinar:

```
ls -la
```

muestra datos en formato largo del archivo oculto.

```
rm nota.vacia
```

borra el archivo `nota.vacia`. Como muchos comandos de Unix/Linux, su nombre deriva de palabras inglesas: `rm` proviene de "remove", eliminar o borrar.

```
rm -i man.txt
```

borrado interactivo, pide confirmación antes de actuar.

```
rm .archivo_oculto; ls -la
```

borra el archivo oculto y verifica listando los archivos del directorio. Los comandos separados por punto y coma (;) se ejecutan secuencialmente, como si estuvieran en líneas diferentes. Esto es diferente del concatenado con `|` donde la salida de un comando va hacia el siguiente; aquí los comandos son independientes, la salida de cada uno se muestra independientemente del otro.

```
echo Mensaje en pantalla
```

muestra en la pantalla la leyenda indicada.

```
echo Este es el archivo mensaje1 > mensaje1
```

direcciona la salida del comando `echo` y graba la leyenda en el archivo `mensaje1`.

```
ls -l mensaje1
```

```
cat mensaje1
```

muestra datos del archivo y verifica su contenido.

```
echo Esta línea es agregada >> mensaje1
```

```
echo Esta es otra línea agregada >> mensaje1
```

```
cat mensaje1
```

redirecciona la salida de `echo` para agregar dos líneas más al archivo ya existente `mensaje1`; verifica el contenido con `cat`.

Diferencia entre operadores de redirección:

```
> sal1 : si el archivo sal1 ya existe lo sobrescribe con el nuevo contenido.
```

```
>> sal1 : si el archivo sal1 ya existe agrega el nuevo contenido al final.
```

```
cat < mensaje1
```

análogo a `cat mensaje1`, pero conceptualmente diferente: se está indicando explícitamente dirigir el contenido del archivo `mensaje1` hacia el comando `cat`. Escribiendo:

```
cat - < mensaje1
```

se obtiene el mismo resultado pero haciendo explícita la "entrada estándar" al comando, indicada por el signo de menos. Si escribimos

```
cat -
```

el comando queda esperando una entrada. Si escribimos algo y digitamos Enter, veremos como el comando lo repite, cumpliendo su función de presentar en pantalla lo que llega a su entrada

estándar, en este caso nuestra digitación en el teclado. Cuando este diálogo le aburra, puede terminar la experiencia digitando Ctrl-C.

Usuarios y grupos

```
who
```

muestra los usuarios que están activos en el sistema (haciendo algo). Indica identificador de usuario, terminal en que está conectado, fecha y hora de ingreso al sistema. Al menos debe aparecer el usuario operador.

```
whoami
```

muestra el nombre del usuario que está operando.

En Unix/Linux cada usuario tiene un nombre de usuario único en el sistema, y está asignado a un grupo principal también único; puede además pertenecer a uno o varios grupos secundarios. Esto puede verse en la salida del comando `id`:

```
victor@uma /tmp $ id
uid=1013(victor) gid=1013(victor)
grupos=1013(victor),20(dialout),27(sudo),46(plugdev),1009(nsk)
```

El usuario `victor` está trabajando en la máquina `uma`, en el directorio `/tmp`; el comando dado es `id`. En la salida, se indica:

```
uid=1013(victor)    número de usuario o UID, 1013 y el nombre de usuario victor.
gid=1013(victor)   número de grupo principal o GID, 1013, también de nombre victor.
grupos=...         la lista de grupos a que pertenece el usuario (número y nombre).
```

Por razones de seguridad, el grupo principal de un usuario es siempre un grupo con el mismo nombre del usuario. Aunque tienen el mismo nombre, son conceptos diferentes: uno es un nombre de usuario, el otro es un nombre de grupo.

```
id jperez
```

proporciona los mismos datos pero para otro usuario del sistema.

```
less /etc/passwd
```

muestra todos los usuarios del sistema, algunos personas reales y otros usuarios definidos por el propio sistema u otros programas.

```
less /etc/group
```

muestra todos los grupos del sistema. Los grupos permiten a varios usuarios compartir permisos y accesos a los mismos datos y programas.

Cambio de contraseña

Un usuario puede cambiar su contraseña con el comando

```
passwd
```

Pide la vieja contraseña y luego la nueva; la nueva contraseña deberá ingresarse dos veces, para evitar posibles errores de digitación.

Caracteres especiales

Las teclas `Ctrl-C` interrumpen la ejecución de un comando. Las teclas `Ctrl-D` indican un fin de ingreso; si se dan en el indicador de comandos, termina la sesión Linux. En Linux no hay un carácter reservado para fin de archivo; `Ctrl-D` simplemente indica la terminación de un flujo de datos.

El caracter `~` (tilde) es un sustituto del directorio propio del usuario. Para el usuario `agarcia`, escribir `~` equivale a escribir `/home/agarcia`, en cualquier comando.

Los caracteres `*` y `$` actúan como comodines: el asterisco equivale a una secuencia arbitraria de caracteres, el signo de pesos a un caracter arbitrario:

```
ls /var/log/*.log
```

muestra todos los nombres de archivo que terminan en `.log`.

```
ls /var/log/syslog.?.gz
```

muestra los nombres de archivo llamados `syslog.?.gz`, donde `$` equivale a cualquier caracter.

Si bien pueden estar permitidos, es mejor evitar el uso de caracteres especiales en los nombres de archivos y directorios; su uso puede resultar problemático mientras no se domine bien la línea de comandos. Se recomienda evitar el uso de los caracteres:

`! | @ # $ % & / () [] { } " ' ^ ` ? : ; , < >`

así como el uso de `-` al principio de un nombre (los comandos lo confundirán con una opción).

Referencias, lecturas complementarias

- Comandos utilizados: `cat` `cd` `date` `echo` `exit` `head` `hostname` `id` `less` `ls` `man` `passwd` `ps` `pwd` `rm` `tail` `touch` `uname` `who` `whoami`
- *Reseña de comandos*, descripción de los comandos utilizados.
- Páginas man: `bash`, páginas de los comandos utilizados.
- Wikipedia, artículos principales: [bash \(Unix shell\)](#).
- Shotts, William. *The Linux Command Line*. Third Internet Edition, 2016. Disponible online: <https://razaoinfo.dl.sourceforge.net/project/linuxcommand/TLCL/16.07/TLCL-16.07.pdf>
Sitio web del autor, con instructivos y referencias: <http://www.linuxcommand.org/index.php>



Copyright: Victor Gonzalez-Barbone.

Esta obra se publicada bajo una Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional.

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.