

Comandos básicos del Shell Bash

El Shell por defecto usado en muchas distribuciones Linux es el Shell bash de GNU. En esta sección se describen las características básicas que están disponibles en el shell bash, tales como las páginas del manual o *man pages*, auto terminación (auto-completion) y como desplegar el contenido de archivos. Se explicará paso a paso como trabajar con archivos y directorios en Linux usando los comandos básicos proveídos por el shell bash. Si usted tiene experiencia previa con los comandos básicos del entorno Linux, puede saltar esta sección y continuar con la siguiente en la que se detallan comandos mas avanzados.

Introducción al Shell

El shell bash de GNU es un programa que provee acceso interactivo a sistemas Linux. Este corre como cualquier programa y normalmente es iniciado en el momento que el usuario inicia sesión (entrada) al sistema. El tipo de shell que inicia el sistema dependerá de la configuración del ID de usuario.

El archivo `/etc/passwd` contiene una lista de todos los usuarios del sistema, así como configuraciones básicas de cada usuario. A continuación, se muestra como luce una entrada del archivo `/etc/passwd`:

```
christine:x:501:501:Christine Bresnahan:/home/christine:/bin/bash
```

Cada entrada tiene siete campos de datos, los cuales están representados o separados por dos puntos. El sistema utiliza los datos de estos campos para asignar características específicas a cada usuario. La mayoría de estas entradas serán discutidas en secciones siguientes, por ahora solo ponga atención al ultimo campo, el cual especifica el programa shell del usuario.

En el archivo de ejemplo `/etc/passwd` anterior, el usuario Christine tiene como programa shell por defecto, el `/bin/bash`. Esto significa que cuando Christine entra al sistema, el programa shell bash es iniciado automáticamente.

Aunque el shell bash es iniciado automáticamente cuando el usuario inicia sesión en el sistema, la interfaz de línea de comandos del shell presentada dependerá de cual método de login haya sido utilizado. Si se utiliza una terminal de consola virtual, el sistema presenta automáticamente el prompt del CLI con lo que ya podría iniciar a escribir comandos. Sin embargo, si usted inicia sesión en el sistema a través del entorno de escritorio gráfico, necesitara iniciar un emulador de terminal grafica para acceder al prompt del shell.

Uso del intérprete de comandos

Después de iniciar un paquete de emulación de terminal o de iniciar sesión a través de una consola virtual de Linux, usted obtiene acceso al prompt del shell. El prompt es la puerta de entrada al shell, es donde usted puede ingresar comandos del shell.

El símbolo por defecto del prompt para el shell bash es el signo de dollar (\$). Este símbolo indica que el shell está esperando que usted ingrese texto. Las diferentes distribuciones de Linux utilizan distintos formatos para el prompt. En el caso de un sistema Linux Ubuntu, el prompt del shell luce de la siguiente manera:

```
christine@server01:$
```

Además de ser el principal punto de entrada al shell, el prompt puede proveer información adicional muy importante. En el ejemplo anterior, el ID del usuario actual, `christine`, es mostrado por el prompt. también se muestra el nombre del sistema, `server01`. Mas adelante aprenderá acerca de otros ítems mostrados por el prompt.

El prompt del shell no es estático. Este puede cambiar para cumplir sus necesidades, en secciones siguientes veremos cómo se configuran las variables ambientes del sistema Linux.

Piense en el prompt del shell como un asistente que le ayuda a interactuar con su sistema Linux, dándole ideas de mucha ayuda y haciéndolo saber cuándo el shell está listo para aceptar nuevos comandos. Otro ítem importante del shell es el manual del bash, visto a continuación.

Interactuando con los man pages.

La mayoría de distribuciones Linux incluyen un manual online que busca información sobre los comandos del shell, así como muchas utilidades GNU que se incluyen en la distribución. Usted debería familiarizarse con el manual, porque es fundamental al trabajar con comandos, especialmente cuando usted necesite investigar como trabajar con varios parámetros desde la línea de comandos.

El comando `man`, da acceso a las páginas del manual almacenadas en un sistema Linux. Al ingresar el comando `man` seguido del nombre de un comando en específico obtenemos como salida la entrada del manual correspondiente a esa utilidad. La figura 4.1 muestra un ejemplo de cómo luce una página del manual para el comando `xterm`. Esta página se obtiene al escribir el comando `man xterm`:

```
XTERM(1)                                X Window System                                XTERM(1)

NAME
    xterm - terminal emulator for X

SYNOPSIS
    xterm [-toolkitoption ...] [-option ...] [shell]

DESCRIPTION
    The xterm program is a terminal emulator for the X Window System. It
    provides DEC VT102/VT220 and selected features from higher-level termi-
    nals such as VT320/VT420/VT520 (VTxxx). It also provides Tektronix
    4014 emulation for programs that cannot use the window system directly.
    If the underlying operating system supports terminal resizing capabili-
    ties (for example, the SIGWINCH signal in systems derived from 4.3bsd),
    xterm will use the facilities to notify programs running in the window
    whenever it is resized.

    The VTxxx and Tektronix 4014 terminals each have their own window so
    that you can edit text in one and look at graphics in the other at the
    same time. To maintain the correct aspect ratio (height/width), Tek-
    tronix graphics will be restricted to the largest box with a 4014's
    aspect ratio that will fit in the window. This box is located in the
    upper left area of the window.

    Although both windows may be displayed at the same time, one of them is
    considered the "active" window for receiving keyboard input and termi-
    nal output. This is the window that contains the text cursor. The
    active window can be chosen through escape sequences, the "VT Options"

Manual page xterm(1) line 1 (press h for help or q to quit)
```

Figura 4.1 paginas del manual para comando xterm

Note que la sección DESCRIPCION del comando `xterm`, muestra una documentación bastante rustica. El manual del `bash` no es una guía paso a paso, mas bien es una guía de referencia rápida. Como usuario nuevo en Linux muy pronto se dará cuenta que los `man pages` no son de mucha ayuda; sin embargo, es importante acostumbrarse a su uso, sobre todo para ver de manera rápida la descripción del comando.

Cuando se utiliza el comando `man` para ver las paginas del manual de un comando en específico, la salida es mostrada con algo llamado pager. El pager es una utilidad que le permite hacer paginación a través del texto mostrado. Esto es, usted utiliza la barra espaciadora para avanzar página a página por el `man page`, o utilizar la tecla Enter para avanzar línea a línea. Adicionalmente puede utilizar las flechas direccionales hacia adelante y hacia atrás.

Al termina de utilizar el `man pages` puede presionar la tecla `q` para salir. Cuando usted sale del `man page`, se vuelve a mostrar el prompt del shell, indicando que el shell está esperando a que se ingresen nuevos comandos.

Las páginas del manual dividen la documentación en secciones separadas. Cada sección tiene una nomenclatura convencional standard, la tabla 4.1 muestra la descripción de cada una de sus secciones:

Sección	Descripción
Name	Muestra el nombre del comando y una breve descripción.
Synopsis	Muestra la sintaxis del comando
Configuration	Provee información de configuración
Description	Describe de manera general un comando
Options	Describe opciones del commando
Exit Status	Define indicadores de salida de un comando
Return value	Describe valores de retorno de un comando
Errors	Provee mensajes de error
Environment	Describe las variables de ambiente utilizadas
Files	Define los archivos utilizados por el comando
Versions	Describe la versión del comando
Conforming To	Provee los estandares seguidos
Notes	Describe material adicional de ayuda del comando
Bugs	Provee la ubicación para reportar los bugs encontrados
Example	Muestra ejemplos de uso del comando
Authors	Provee información sobre desarrolladores del comando
Copyright	Define el estado del copyright del código fuente del comando
See Also	Referencia a comandos similares

Tabla 4.1 Secciones de convención de nombres de los `man pages`

No todos los man pages tienen todas las secciones antes descritas. También, hay algunos comandos que muestran otras secciones no descritas en la tabla 4.1

Adicionalmente a estas secciones, hay áreas de secciones dentro del man page. Cada área tiene asignado un número, desde el 1 hasta el 9. La tabla 4.2 lista las áreas de sección de los man pages:

Sección	Descripción
1	Programas ejecutables o comandos del shell
2	Llamadas al sistema
3	Llamadas a librería
4	Archivos especiales
5	Formatos de archive
6	Juegos
7	Algunas convenciones y misceláneas
8	Super usuario y comandos de administración del sistema
9	Rutinas del kernel

Tabla 4.2 Areas de sección de los man pages

Normalmente, el comando man provee el contenido del número más bajo. Por ejemplo, si observa la figura 4.1 donde se muestra la salida del comando `man xterm`, note que en la parte superior derecha y la parte superior izquierda el nombre del comando está acompañado del número del área de sección del comando 1. Esto significa que el man page mostrado proviene del área 1(programas ejecutables y comandos shell)

Ocasionalmente, el man page de un comando puede tener múltiples áreas. Por ejemplo, el comando llamado `hostname` contiene varias secciones. Para ver la página deseada la sintaxis del comando es `man seccion# comando`. Para ver man page en la sección 1, escriba `man 1 hostname`. Para ver el man page en la sección 7, escriba `man 7 hostname`.

Los man pages no son la única referencia. También hay páginas de información llamadas las info pages. Puede explorar las info pages escribiendo el comando `info info`.

Adicionalmente, la mayoría de comandos aceptan la opción `-help`. Por ejemplo, si escribe la opción `hostname -help` mostrará una ayuda rápida del comando. Para más información sobre el comando `help` utilice el `help help`.

Obviamente hay muchos recursos de referencia de utilidad. Sin embargo, muchos conceptos básicos del Shell requieren una explicación detallada. En la siguiente sección, se explora el sistema de archivos de Linux.

Navegando por el sistema de archivos

Cuando usted inicia sesión y entra al prompt de comandos del Shell, por defecto es ubicado en su directorio home. Por lo general es de su interés explorar otras áreas del sistema Linux además del directorio home personal. Esta sección describe como hacer esto usando comandos del Shell. Para iniciar, debe hacer un recorrido por el sistema de archivos.

El sistema de archivos de Linux

Si usted es nuevo en el uso del sistema de archivos de Linux, puede ser que le confunda la forma en que se referencian los archivos y directorios, especialmente si está acostumbrado a los sistemas operativos Microsoft Windows. Antes de explorar el sistema de Linux, recordaremos la forma en que Windows explora el sistema de directorios.

La primera diferencia que notatara es que Linux no utiliza letras de unidades en las rutas de directorio. En el mundo de Windows, la unidad física instalada en la computadora determina la ruta al archivo. Windows asigna una letra a cada unidad de disco físico, y cada unidad contiene su propia estructura de directorios.

Por ejemplo, la siguiente ruta de Windows le puede ser muy familiar:

```
c:\Users\Rich\Documents\test.doc
```

La ruta de Windows indica exactamente que partición del disco contiene el archivo `test.doc`. por ejemplo, si guarda el archivo `test.doc` en una unidad USB, mostrada en el sistema como unidad J, la ruta al archivo sería `J:\test.doc`. esta indica que el archivo está localizado en la raíz del disco que tiene asignada la letra J.

El método anterior no es el utilizado por Linux. Linux almacena los archivos dentro de una estructura única de directorio, llamada directorio virtual. El directorio virtual contiene la ruta de los archivos de todos los dispositivos de almacenamiento instalados en el equipo, unido en una sola estructura de directorios.

La estructura de directorio virtual de Linux contiene un solo directorio base, llamado root. Los directorios y archivos debajo del root (/) se enumeran en función de la ruta definida para llegar a ellos, similar a la manera en que lo hace Windows.

En Linux, la ruta a un archivo luce de la siguiente manera:

```
/home/Rich/Documents/test.doc
```

Esta indica que el archivo `test.doc` está dentro del directorio `Documents`, dentro del directorio `Rich`, el cual está contenido en el directorio `/home`. Tena en cuenta que la ruta no muestra ninguna información en relación al disco físico que almacena el archivo.

La parte clave de los directorios virtuales en Linux es como maneja cada dispositivo de almacenamiento. El primer disco instalado en un sistema Linux se llama unidad root o raíz. Linux hace que los archivos y directorios aparezcan dentro de estos directorios raíz, llamados puntos de montaje.

A veces los archivos del sistema son almacenados en unidades raíz. Los archivos de usuario generalmente se almacenan en una sola unidad o unidades separadas, como se muestra en la figura 4.2:

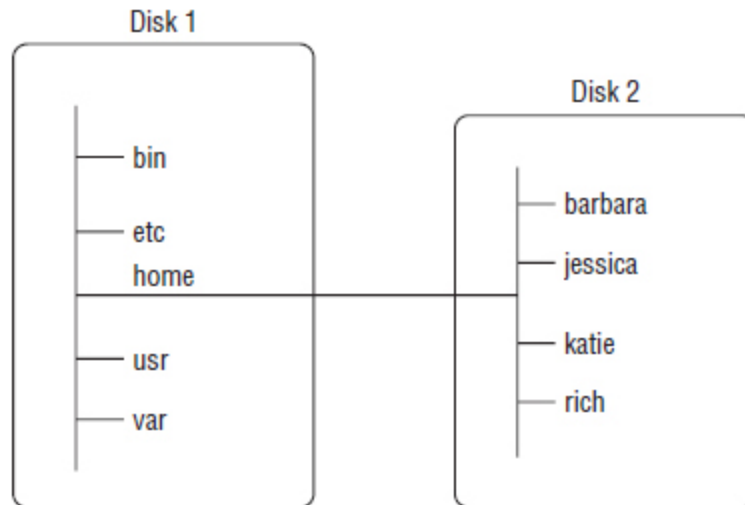


Figura 4.2 Estructura de archivos en Linux

La figura 4.2 muestra dos discos duros de un mismo equipo. Un disco duro es asociado con la raíz del directorio virtual (se indica por el carácter / del disco 2). El otro disco puede ser montado en cualquier lugar de la estructura de directorio virtual. En este ejemplo, el segundo disco se monta en la ubicación `/home`, que es donde se encuentran los directorios de usuario final.

La estructura del sistema de archivos de Linux, originalmente evolucionó de la estructura de archivos de Unix. En un sistema de archivos Linux, los directorios tienen funciones predeterminadas. La tabla 4.3 lista algunos de los directorios de nivel superior más comunes, así como su función:

Directorio	Uso
/	Raíz del directorio virtual, normalmente no aloja ningún archivo.
/bin	Directorio de binarios, donde son almacenadas muchas utilidades GNU de nivel de usuario.
/boot	Directorio de inicio, donde son almacenados los archivos de arranque del sistema
/dev	Directorio de dispositivos, donde Linux crea los archivos de tipo dispositivo
/etc	Archivos de configuración de sistema
/home	Directorio home, donde Linux crea los directorios de usuario
/lib	Directorio de librerías, donde son almacenados los archivos de librería de aplicación y sistema.
/media	Directorio de media, un directorio para manejo de puntos de montaje de los medios de almacenamiento

/mnt	Similar al anterior
/opt	Directorio opcional, muy usado para almacenar software de terceros.
/proc	Directorio de procesos, donde se almacena el hardware actual y la información de procesos
/root	Directorio del usuario root o administrador (no confundir con el directorio root)
/sbin	Directorio de binarios del sistema, donde son almacenadas muchas utilidades GNU de nivel administrad
/run	Directorio run, donde se mantienen datos de ejecución de programas durante la operación del sistema
/srv	Directorio de servicios, donde se almacenan los archivos de servicios locales
/sys	Directorio de sistema, donde se almacenan archivos de información del hardware del sistema
/tmp	Directorio temporal, donde son creados y destruidos archivos temporales
/usr	Directorio de binarios de usuario, donde se almacenan utilidades GNU de nivel de usuario.
/var	Directorio variable, para almacenamiento de archivos que cambian con frecuencia, tales como los logs

Tabla 4.3 Nombres de los directorios Linux

Los nombres de directorios de Linux están basados en el estándar FHS (Filesystem Hierarchy Standard). Muchas distribuciones Linux cumplen con el FHS. Por lo tanto, cualquier distribución Linux que cumpla el estándar debería ser muy familiar.

Cuando inicia sesión en el sistema y llega al prompt del Shell, su sesión es ubicada en su directorio personal. Su directorio home es un directorio único asignado a su cuenta de usuario. Cuando se crea una cuenta de usuario, el sistema asigna un directorio home a la cuenta de usuario.

Al igual que Windows se puede navegar por el sistema de archivos utilizando una aplicación gráfica. Sin embargo, para poder moverse por el directorio virtual desde un prompt del Shell, debe aprender a utilizar el comando `cd` (change directory).

Navegando por los directorios

Para mover su sesión del Shell desde un directorio a otro se utiliza el comando de cambio de directorio `cd`. La sintaxis del comando `cd` es bastante simple: `cd <destino>`.

El comando `cd` puede tomar solamente un parámetro, `<destino>`, el cual especifica el nombre del directorio al que quiere ir. Si no se especifica el nombre del directorio, el comando lo lleva directamente al directorio `home`.

El parámetro de destino se puede expresar utilizando dos métodos diferentes. El primero es utilizando una referencia de directorio absoluta. El segundo método es usando una referencia de directorio relativa.

Las siguientes secciones describen estos dos métodos. Es importante comprender las diferencias entre estos dos métodos a medida que avanza en el uso de Linux.

1. Referencias de directorio absoluto:

Usted puede referenciar un nombre de directorio dentro del sistema de directorio virtual utilizando una referencia de directorio absoluto. Esta define donde exactamente está el directorio en la estructura de directorio virtual. Iniciando en el directorio root. La referencia de directorio absoluto corresponde al nombre completo del directorio.

Una referencia de directorio absoluto siempre inicia con el carácter slash (/), que indica la raíz del sistema de directorio virtual. Por lo tanto, para hacer referencia a los binarios del usuario, contenidos dentro del directorio `bin` y este a su vez almacenado en el directorio `usr`, usará una referencia de directorio absoluta:

```
/usr/bin
```

Utilizando la referencia de directorio absoluta no hay ninguna duda hacia donde se dirige.

Para moverse a una dirección específica utilizando la referencia absoluta, basta con especificar el nombre completo de la ruta como parámetro del comando `cd`:

```
christine@server01:~$ cd /usr/bin
christine@server01:/usr/bin$
```

En el ejemplo anterior observe que en la primera línea se observa el uso de virgulilla (~). Después de cambiar de directorio la virgulilla es reemplazada por `/usr/bin`. Es esta la manera en que el prompt del CLI puede ayudarlo a ubicarse en el sistema de archivos. El carácter `~` indica que la sesión del Shell se encuentra en su directorio home. Después

de salir de su directorio home, si el prompt ha sido configurado correctamente, se muestra la referencia absoluta al directorio, es decir el directorio actual de trabajo.

Si su prompt no ha sido configurado para mostrar la ubicación del directorio absoluto, puede utilizar un comando del Shell para mostrarla. el comando `pwd` muestra la ubicación de trabajo actual de la sesión del shell, el cual es llamado directorio de trabajo actual. A continuación, se muestra un ejemplo del uso del comando `pwd`:

```
christine@server01:/usr/bin$ pwd
/usr/bin
christine@server01:/usr/bin$
```

Usted puede moverse hacia cualquier directorio dentro de la estructura de directorio virtual de Linux desde cualquier nivel, utilizando la referencia de directorio absoluta.

```
christine@server01:/usr/bin$ cd /var/log
christine@server01:/var/log$
christine@server01:/var/log$ pwd
/var/log
christine@server01:/var/log$
```

También puede regresar rápidamente a su directorio home desde cualquier lugar de la estructura de directorios usando el comando `cd`:

```
christine@server01:/var/log$ cd
christine@server01:~$
christine@server01:~$ pwd
/home/christine
christine@server01:~$
```

Sin embargo, si usted solamente está trabajando dentro de su propio directorio home, utilizar referencias absolutas puede ser tedioso. Por ejemplo, si usted ya está dentro del directorio `/home/christine`, evidentemente es algo engorroso tener que escribir el comando con su ruta absoluta:

```
cd /home/christine/Documents
```

Escribir la ruta completa solo para llegar a un nivel superior a su ubicación actual, afortunadamente existe una solución mas simple, veamos a continuación.

2. Referencias de directorio relativo:

Las referencias de directorio relativo le permiten especificar una referencia de directorio de destino relativa a su ubicación actual. Una referencia de directorio relativa no inicia con el carácter slash (/).

En cambio, una referencia de directorio relativa comienza con un nombre de directorio o con un carácter especial. Por ejemplo, si usted está en su directorio `home` y se quiere mover al subdirectorio `Documents` ubicado dentro de su directorio `home`, puede utilizar el comando `cd` con una referencia de directorio relativa:

```
christine@server01:~$ pwd
/home/christine
christine@server01:~$
christine@server01:~$ cd Documents
christine@server01:~/Documents$ pwd
/home/christine/Documents
christine@server01:~/Documents$
```

En el ejemplo anterior, tenga en cuenta que no hay barra diagonal (/). En su lugar se utiliza una referencia de directorio relativa, el directorio de trabajo actual `/home/christine` fue cambiado por `/home/christine/Documents`, con mucho menos caracteres que escribir.

Observe también, que si el prompt está configurado para mostrar el directorio de trabajo actual mantiene el carácter `~` en pantalla. Se muestra que el directorio de trabajo actual está bajo el directorio `home` del usuario.

Usted puede utilizar referencia de directorio relativa en cualquier directorio que contenga subdirectorios utilizando el comando `cd`. También puede utilizar un carácter especial para indicar una ubicación de directorio relativa.

Los dos caracteres especiales utilizados para referencia de directorio relativa son:

- El punto único (`.`) para representar el directorio actual.
- El punto doble (`..`) para representar el directorio padre.

Usted puede utilizar el punto único, sin embargo, no tiene sentido usarlo con el comando `cd`. Mas adelante veremos cómo utilizar efectivamente otra forma del punto único para referencias relativas a directorios.

El punto doble es extremadamente útil cuando se trata de atravesar jerarquías de directorios. Por ejemplo, si está ubicado en el directorio `Documents` debajo de su directorio `home` y necesita ir a su directorio `Downloads`, también ubicado dentro de su directorio `home`, puede hacer lo siguiente:

```
christine@server01:~/Documents$ pwd
/home/christine/Documents
christine@server01:~/Documents$ cd ../Downloads
christine@server01:~/Downloads$ pwd
/home/christine/Downloads
christine@server01:~/Downloads$
```

El carácter punto doble lo sube un nivel, ubicándolo en su directorio `home`, entonces la parte `/Downloads` del comando lo lleva de regreso al directorio `Downloads`. Puede utilizar los caracteres de punto doble que sean necesarios. Por ejemplo, si usted está en su directorio `home` (`/home/christine`) y quiere ir al directorio `/etc`, puede utilizar el comando `cd` siguiente:

```
christine@server01:~$ cd ../../etc
christine@server01:/etc$ pwd
/etc
christine@server01:/etc$
```

Por supuesto, en un caso como este, puede ser más fácil utilizar la referencia absoluta `/etc`. Por lo tanto, use una referencia de directorio relativa solamente en caso de tener sentido hacerlo.

Ahora que ha aprendido como recorrer el sistema de directorios, puede comenzar a explorar los directorios de nivel superior mostrados en la tabla 4.3. En la siguiente sección se muestra el proceso de buscar y mostrar archivos en una estructura de directorio virtual.

Listando archivos y Directorios

Para ver que archivos están disponibles un directorio dado, se utiliza el comando `ls`. Esta sección describe el comando `ls`, así como las opciones disponibles para darle formato a su salida.

Visualizando una lista básica

El comando `ls` en su forma más básica muestra los archivos y directorios del directorio actual de trabajo:

```
$ ls
Desktop    Downloads      Music          Pictures    Templates    Videos
Documents  examples.desktop  my_script    Public      test_file
$
```

Observe que el comando produce un listado ordenado alfabéticamente (en columnas en lugar de filas). Si utiliza un emulador de terminal que admite colores, el comando `ls` puede mostrar los diferentes tipos de entradas en colores distintos. La variable ambiente `LS_COLORS` controla esta característica (más adelante estudiaremos las variables de ambiente). Las

diferentes distribuciones de Linux configuran esta variable ambiente dependiendo de las capacidades gráficas del emulador de terminal.

Si no cuenta con un emulador de terminales que soporte colores, puede utilizar el parámetro `-F`, de esta manera diferenciará los archivos de los directorios. El uso del parámetro `-F` produce el siguiente resultado:

```
$ ls -F
Desktop/  Downloads/      Music/      Pictures/ Templates/ Videos/
Documents/ examples.desktop my_script* Public/    test_file
$
```

El parámetro `-F` marca los directorios con el signo `/`, para ayudar a identificarlos en el listado. Del mismo modo marca los archivos ejecutables con el signo asterisco (`*`), para ayudar a encontrar más fácilmente los archivos que pueden ejecutarse en el sistema.

La salida simple del comando `ls` puede ser engañosa. Muestra los archivos y directorios del directorio actual, pero no necesariamente todos. Linux, a menudo utiliza archivos ocultos para almacenar información de configuración. Los archivos ocultos son los archivos cuyo nombre inicia con punto (`.`). Estos archivos no aparecen en la lista predeterminada. Por lo tanto, son llamados archivos ocultos.

Para mostrar los archivos ocultos junto a los archivos y directorios normales, utilice el parámetro `-a`. Aquí hay un ejemplo del uso del comando `ls` con el parámetro `-a`:

```
$ ls -a
.          .compiz    examples.desktop Music      test_file
..         .config    .gconf          my_script Videos
.bash_history Desktop    .gstreamer-0.10 Pictures   .Xauthority
.bash_logout .dmrc      .ICEauthority   .profile  .xsession-errors
.bashrc      Documents  .local          Public     .xsession-errors.old
.cache       Downloads  .mozilla        Templates
$
```

Vea que ahora se muestran todos los archivos que inician con un punto (archivos ocultos). Observe que hay tres archivos que inician con `.bash`. Estos son archivos ocultos utilizados en el entorno del Shell `bash`. Estas características se verán en detalle en secciones siguientes.

El parámetro `-R` es otra opción que puede utilizar el comando `ls`. Llamado la opción recursiva, esta opción muestra los archivos que están contenidos dentro de los subdirectorios del directorio actual. Si tiene muchos subdirectorios, esta lista puede ser muy extensa. A continuación, se muestra un ejemplo sencillo de la salida producida por el parámetro `-R`, se agrega la opciones `-F` para ayudarlo a diferenciar los tipos de archivos:

```
$ ls -F -R
.:
Desktop/  Downloads/      Music/      Pictures/  Templates/  Videos/
Documents/ examples.desktop my_script*  Public/    test_file

./Desktop:

./Documents:

./Downloads:

./Music:
ILoveLinux.mp3*

./Pictures:

./Public:

./Templates:

./Videos:
$
```

Observe que el parámetro `-F` muestra el contenido del directorio actual, el cual son los archivos del directorio `home` del usuario de ejemplos anteriores. También muestra los

subdirectorios y su contenido. El único subdirectorio que contiene un archivo es el subdirectorio `Music`, que contiene el archivo ejecutable `ILoveLinux.mp3`

En el ejemplo anterior no hay subdirectorios dentro de los subdirectorios. De haber más subdirectorios, el parámetro `-R` los habría mostrado también. Como puede ver, para estructuras de directorio grandes, la salida puede convertirse en una lista bastante extensa.

Visualizando una lista larga

En los listados básicos, el comando `ls` no muestra mucha información de los archivos. Para mostrar información adicional existe otro parámetro muy popular, la opción `-l`. El parámetro `-l` produce un formato de listado largo que provee más información acerca de cada archivo dentro del directorio:

```
$ ls -l
total 48
drwxr-xr-x 2 christine christine 4096 Apr 22 20:37 Desktop
drwxr-xr-x 2 christine christine 4096 Apr 22 20:37 Documents
drwxr-xr-x 2 christine christine 4096 Apr 22 20:37 Downloads
-rw-r--r-- 1 christine christine 8980 Apr 22 13:36 examples.desktop
-rw-rw-r-- 1 christine christine    0 May 21 13:44 fall
-rw-rw-r-- 1 christine christine    0 May 21 13:44 fell
-rw-rw-r-- 1 christine christine    0 May 21 13:44 fill
-rw-rw-r-- 1 christine christine    0 May 21 13:44 full
drwxr-xr-x 2 christine christine 4096 May 21 11:39 Music
-rw-rw-r-- 1 christine christine    0 May 21 13:25 my_file
-rw-rw-r-- 1 christine christine    0 May 21 13:25 my_script
-rwxrw-r-- 1 christine christine   54 May 21 11:26 my_script
-rw-rw-r-- 1 christine christine    0 May 21 13:42 new_file
drwxr-xr-x 2 christine christine 4096 Apr 22 20:37 Pictures
drwxr-xr-x 2 christine christine 4096 Apr 22 20:37 Public
drwxr-xr-x 2 christine christine 4096 Apr 22 20:37 Templates
-rw-rw-r-- 1 christine christine    0 May 21 11:28 test_file
drwxr-xr-x 2 christine christine 4096 Apr 22 20:37 Videos
$
```

El formato de listado largo muestra una línea por cada archivo y subdirectorio. Además del nombre del archivo, el listado muestra información adicional de mucha utilidad. La primera línea de la salida imprime el número total de bloques que contiene el directorio. A continuación, van las líneas que contienen la información detallada de cada archivo o subdirectorio.

- El tipo de archivo: (d) directorio, (-) archivo, (l) archivo tipo enlace, (c) dispositivo tipo carácter o (b) dispositivo tipo bloque.
- Los permisos del archivo
- El número de enlaces duros o archivos de enlace físico.
- El nombre del usuario propietario del archivo
- El nombre del grupo primario del archivo
- El tamaño del archivo en bytes
- La última vez que se modificó el archivo
- El nombre del archivo o el nombre del directorio

El parámetro `-l` es una herramienta muy poderosa a tener en cuenta. Con este parámetro, puede ver la mayoría de la información que usted necesita de un archivo.

El comando `ls` tiene varios otros parámetros que le pueden ser de mucha utilidad para la administración de un sistema Linux. Si escribe `man ls` en el prompt del Shell, podrá ver los parámetros disponibles, pruebe algunos de ellos para ver la forma en que afecta su salida.

No olvide que los parámetros se pueden combinar. Una combinación de mucha utilidad es el conjunto de parámetros `-alF`.

Filtrando la salida del listado de directorio.

Como ha podido observar en los ejemplos anteriores, el comando `ls` despliega todos los archivos que no inician con punto (archivos ocultos). Sin embargo, a veces esto puede ser demasiado, especialmente cuando lo que interesa es buscar información sobre unos cuantos archivos.

Afortunadamente, el comando `ls` soporta parámetros que filtran la salida, los filtros son normalmente utilizados para determinar que archivos o directorios se deben mostrar por pantalla.

El filtro funciona de manera simple, pasamos la cadena de coincidencia, el filtro se puede incluir después de cualquier parámetro de línea de comandos que desee utilizar:

```
$ ls -l my_script
-rwxrw-r-- 1 christine christine 54 May 21 11:26 my_script
$
```

Cuando se especifica la cadena que se desea filtrar, el comando `ls` muestra información del archivo cuyo nombre coincide con esa cadena. A veces, es posible que no sepa el nombre exacto del archivo que desea buscar.

El comando `ls` también reconoce los patrones “comodín estándar” y los usa para unir patrones dentro del filtro.

- Un signo de interrogación (?) para representar un carácter.
- Un asterisco (*) para representar cualquier número de caracteres.

El signo de interrogación puede utilizarse para reemplazar un carácter en cualquier lugar de la cadena de filtro. Por ejemplo:

```
$ ls -l my_scr?pt
-rw-rw-r-- 1 christine christine  0 May 21 13:25 my_scrapt
-rwxrw-r-- 1 christine christine 54 May 21 11:26 my_script
$
```

El filtro `my_scr?pt` coincide con dos archivos del directorio. Del mismo modo, el asterisco puede utilizarse para hacer coincidir cero o más caracteres.

```
$ ls -l my*
-rw-rw-r-- 1 christine christine  0 May 21 13:25 my_file
-rw-rw-r-- 1 christine christine  0 May 21 13:25 my_scrapt
-rwxrw-r-- 1 christine christine 54 May 21 11:26 my_script
$
```

Utilizando el asterisco se encuentran 3 archivos distintos, todos iniciando con `my`. Al igual que el signo de interrogación, usted puede ubicar el signo asterisco en cualquier lugar de la cadena:

```
$ ls -l my_s*t
-rw-rw-r-- 1 christine christine  0 May 21 13:25 my_script
-rwxrw-r-- 1 christine christine 54 May 21 11:26 my_script
$
```

Al uso del asterisco y del signo de interrogación dentro del Shell se le conoce como file globbing. El File globbing es el proceso de buscar patrones utilizando comodines. Los comodines son formalmente llamados comodines de metacaracter. A parte del asterisco y el signo de interrogación, usted puede utilizar otros comodines de metacaracter para file globbing. En el ejemplo siguiente se usan corchetes:

```
$ ls -l my_scr[ai]pt
-rw-rw-r-- 1 christine christine  0 May 21 13:25 my_script
-rwxrw-r-- 1 christine christine 54 May 21 11:26 my_script
$
```

En este ejemplo, dentro de los corchetes se colocan dos caracteres, cada uno reemplaza una opción de búsqueda. Los corchetes representan la posición de un solo carácter lo cual brinda muchas opciones de búsqueda. Puede especificar un rango de caracteres, por ejemplo, un rango alfabético `[a – i]`:

```
$ ls -l f[a-i]ll
-rw-rw-r-- 1 christine christine 0 May 21 13:44 fall
-rw-rw-r-- 1 christine christine 0 May 21 13:44 fell
-rw-rw-r-- 1 christine christine 0 May 21 13:44 fill
$
```

Por otro lado, puede utilizarse el carácter signo de exclamación para indicar exclusión de coincidencia en el patrón de búsqueda:

```
$ ls -l f[!a]ll
-rw-rw-r-- 1 christine christine 0 May 21 13:44 fell
-rw-rw-r-- 1 christine christine 0 May 21 13:44 fill
-rw-rw-r-- 1 christine christine 0 May 21 13:44 full
$
```

El file globbing es una característica de búsqueda de archivos muy potente. Además del `ls`, puede ser utilizada con otros comandos del Shell.

Bibliografía:

Richard Blum, Christine Bresnahan. (January 2015). Linux Command Line and Shell Scripting Bible. Indianapolis, IN 46256, United States: Wiley.

Rob Kennedy. (Noviembre 2018). Essential Linux Command Line. United States: Kennedy Projects .