

Linux®

Apuntes para Profesionales

Chapter 1: Getting started with GNU

Section 1.1: Useful shortcuts

Using The Terminal

The examples in this document assume that you are using a POSIX-compliant (such as Bash) shell.

Large portions of GNU/Linux functionality are achieved using the terminal. Most distributions provide terminal emulators that allow users to interact with a shell from their desktop environment. A line interpreter that executes user inputted commands, **Bash** (Bourne Again Shell) is a common among many Linux distributions, and is the default shell for many.

These shortcuts will work if you are using **Bash** with the emacs keybindings (set by default).

Open terminal

• **Ctrl + Alt + T** or **Super + T**

Cursor movement

- **Ctrl + A** Go to the beginning of the line you are currently typing on.
- **Ctrl + E** Go to the end of the line you are currently typing on.
- **Ctrl + B** Move between the beginning of the line and the current position.
- **Ctrl + F** Move cursor forward one word on the current line.
- **Alt + B** Move cursor backward one word on the current line.
- **Ctrl + P** Move cursor forward one character on the current line.
- **Ctrl + N** Move cursor backward one character on the current line.

Text manipulation

- **Ctrl + U** Cut the line from the current position to the beginning of the line.
- **Ctrl + K** Cut the line from the current position to the end of the line.
- **Ctrl + W** Cut the line from the beginning of the line, cut the entire line.
- **Ctrl + Y** Paste the last thing from the clipboard that you cut recently.
- **Ctrl + R** Delete the word before the cursor, adding it to the clipboard.
- **Ctrl + Z** Delete the last thing from the clipboard that you cut recently.
- **Ctrl + Y** Paste the last thing from the clipboard that you cut recently.
- **Ctrl + X** Swap the last two words before the cursor.
- **Alt + U** Make lowercase from cursor to end of word.
- **Alt + S** Make uppercase from cursor to end of word.
- **Alt + C** Capitalize to end of word starting at cursor (whole word).
- **Alt + D** Delete to end of word written in previous command.
- **Alt + F** Prints the last word written in previous command.
- **Alt + B** Swap the last two characters before the cursor.
- **Ctrl + T** Swap the last two characters before the cursor.

History access

- **Ctrl + R** Lets you search through previously used commands.
- **Ctrl + S** Lets you search through previously used commands.
- **Ctrl + J** Lets you copy current matched command to command line without running it, allowing you to

Chapter 2: Detecting Linux distribution name and version

Section 2.1: Detect what debian-based distribution you are working in

Just execute **lsb_release -a**.

On Debian:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description: Debian GNU/Linux testing (stretch)
Release: testing
Codename: stretch
```

On Ubuntu:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 14.04.4 LTS
Release: 14.04
Codename: trusty
```

In case when you don't have **lsb_release** installed you may want to try some guessing, for example, there is a file **/etc/issue** that often contains distribution name. For example, on Ubuntu:

```
$ cat /etc/issue
Ubuntu 12.04.3 LTS (12.10)
```

Don't use file **/etc/debian_version** because its contents do not match distribution name.

Note that this will also work on non-Debian-family distributions like Fedora, Arch, or openSUSE — but that **lsb_release** may not be installed.

Section 2.2: Detect what systemd-based distribution you are using

This method will work on modern versions of Arch, CentOS, CoreOS, Debian, Fedora, Mageia, openSUSE, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Ubuntu, and others. This wide applicability makes it an ideal first approach, with fallback to other methods if you need to also identify older systems.

Look at **fedora-release**. In specific, look at variables **NAME**, **VERSION_ID**, **VERSION_TS**, and **PRETTY_NAME**.

On Fedora, this file might look like:

```
NAME=Fedora
VERSION_ID=24 (Workstation Edition)
24-Fedora
VERSION_TS=24
PRETTY_NAME=Fedora 24 (Workstation Edition)
ANSI_COLOR="0;34"
```

Linux® Notes for Professionals

Chapter 5: Check Disk Space

Section 5.1: Investigate Directories For Disk Usage

Sometimes it may be required to find out which directory consuming how much disk space especially when you are used **df -h** and realized your available disk space is low.

du

du command summarizes disk usage of the set of files, recursively for directories. It's often uses with **-sh** options:

```
-s, --summarize
    Display only a total for each argument
-h, --human-readable
    Print sizes in human readable format (e.g., 1K 234M 2G)
```

For summarizing disk usages of the files in the current directory we use:

```
du -sh *
```

Example output:

```
37K  documents
20M  downloads
4.0K  music
72K  pictures
4.0K  public
4.0K  templates
4.0K  videos
```

We can also include hidden files with using:

```
du -sh . | grep *
```

Example output:

```
4.3M  .atom
4.0K  .bash_history
4.0K  .bash_logout
350M  .bashrc
190M  .cache
12M  .config
4.0K  .dove
44K  .dove
60K  .dove
52M  .gnome
28K  .gnome-2.8
4.0K  .gnome
8.3M  .ICEauthority
8.0K  .local
40K  .nautilus
36K  .nvc
40K  .nvc
```

Linux® Notes for Professionals

Traducido por:

ortegag

50+ páginas

de consejos y trucos profesionales

GoalKicker.com
Free Programming Books

Descargo de responsabilidad
Este es un libro gratuito no oficial creado con fines educativos y no está afiliado a ningún grupo o empresa oficial de Linux®. Todas las marcas comerciales y marcas registradas son propiedad de sus respectivos dueños.

Contenidos

Acerca de	1
Capítulo 1: Introducción a GNU/Linux.....	2
Sección 1.1: Atajos útiles	2
Sección 1.2: Comandos de gestión de archivos.....	3
Sección 1.3: Hola Mundo	5
Sección 1.4: Utilidades básicas de Linux.....	5
Sección 1.5: Búsqueda de archivos por patrones en el nombre/contenido.....	6
Sección 1.6: Manipulación de archivos.....	7
Sección 1.7: Detalles del archivo/directorio.....	8
Capítulo 2: Detección del nombre y la versión de la distribución Linux.....	10
Sección 2.1: Detecte en qué distribución basada en Debian está trabajando.....	10
Sección 2.2: Detectar qué distribución basada en systemd está utilizando.....	10
Sección 2.3: Detectar en qué distribución RHEL / CentOS / Fedora está trabajando	11
Sección 2.4: Uname - Imprime información sobre el sistema actual.....	11
Sección 2.5: Detectar información básica sobre su distribución.....	12
Sección 2.6: Utilizar GNU coreutils	12
Sección 2.7: Encuentre el nombre y el número de versión de su sistema operativo Linux (tanto debian como rpm).....	12
Capítulo 3: Obtener información sobre un núcleo Linux en ejecución	14
Sección 3.1: Obtener detalles del núcleo de Linux	14
Capítulo 4: Shell.....	15
Sección 4.1: Cambiar el shell por defecto.....	15
Sección 4.2: Utilidades básicas del shell	16
Sección 4.3: Cree su propio alias de comando.....	17
Sección 4.4: Localiza un archivo en tu sistema	17
Capítulo 5: Comprobar el espacio en disco	18
Sección 5.1: Investigar el uso del disco en los directorios.....	18
Sección 5.2: Comprobar el espacio en disco.....	20
Capítulo 6: Obtener información del sistema	22
Sección 6.1: Estadísticas sobre CPU, Memoria, Red y Disco (operaciones de E/S)	22
Sección 6.2: Uso de herramientas como lscpu y lshw	22
Sección 6.3: Lista de hardware	23
Sección 6.4: Buscar información sobre el modelo y la velocidad de la CPU	24
Sección 6.5: Supervisión de procesos y recopilación de información.....	25
Capítulo 7: comando ls.....	26
Sección 7.1: Opciones del comando ls.....	26
Sección 7.2: Comando ls con las opciones más utilizadas	26
Capítulo 8: Compresión de archivos con el comando ‘tar’	28
Sección 8.1: Comprimir una carpeta	28

Sección 8.2: Extraer una carpeta de un archivo.....	28
Sección 8.3: Listar el contenido de un archivo.....	28
Sección 8.4: Listar el contenido del archivo.....	29
Sección 8.5: Comprimir y excluir una o varias carpetas	29
Sección 8.6: Componentes principales de la banda.....	29
Capítulo 9: Servicios.....	30
Sección 9.1: Lista de servicios en ejecución en Ubuntu	30
Sección 9.2: Gestión de servicios Systemd.....	30
Capítulo 10: Gestión de servicios.....	31
Sección 10.1: Diagnosticar un problema con un servicio	31
Sección 10.2: Iniciar y detener servicios.....	31
Sección 10.3: Obtener el estado de un servicio	31
Capítulo 11: Modificación de usuarios	33
Sección 11.1: Establecer su propia contraseña	33
Sección 11.2: Establecer la contraseña de otro usuario.....	33
Sección 11.3: Añadir un usuario	33
Sección 11.4: Eliminar un usuario.....	33
Sección 11.5: Eliminar un usuario y su carpeta de inicio	33
Sección 11.6: Listado de grupos a los que pertenece el usuario actual.....	33
Sección 11.7: Listado de grupos a los que pertenece un usuario.....	33
Capítulo 12: LAMP Stack.....	34
Sección 12.1: Instalación de LAMP en Arch Linux.....	34
Sección 12.2: Instalación de LAMP en Ubuntu.....	35
Sección 12.3: Instalación de LAMP en CentOS	35
Capítulo 13: comando tee.....	37
Sección 13.1: Escribir la salida a stdout, y también a un archivo.....	37
Sección 13.2: Escribir la salida desde el medio de una cadena de tuberías a un archivo y pasarlo de nuevo a la tubería	37
Sección 13.3: Escribir la salida en varios archivos	37
Sección 13.4: Indica al comando tee que añada al archivo	37
Capítulo 14: Secure Shell (SSH).....	38
Sección 14.1: Conectarse a un servidor remoto.....	38
Sección 14.2: Instalación del paquete OpenSSH.....	38
Sección 14.3: Configurar un servidor SSH para que acepte conexiones	38
Sección 14.4: Conexión sin contraseña (mediante un par de claves).....	39
Sección 14.5: Generar clave pública y privada	39
Sección 14.6: Desactivar el servicio ssh	39
Capítulo 15: SCP	40
Sección 15.1: Copia segura	40
Sección 15.2: Uso básico	40
Capítulo 16: GnuPG (GPG).....	41

Sección 16.1: Exportar la clave pública.....	41
Sección 16.2: Crear y utilizar rápidamente una clave GnuPG.....	41
Capítulo 17: Configuración de red.....	42
Sección 17.1: Resolución DNS local	42
Sección 17.2: Configurar servidores DNS para la resolución de nombres de dominio.....	42
Sección 17.3: Ver y manipular rutas.....	42
Sección 17.4: Configurar un nombre de host para otro sistema de la red	43
Sección 17.5: Detalles de la interfaz.....	43
Sección 17.6: Añadir IP a una interfaz	45
Capítulo 18: Midnight Commander.....	46
Sección 18.1: Teclas de función del Midnight Commander en modo navegación	46
Sección 18.2: Teclas de función de Midnight Commander en modo de edición de archivos.....	46
Capítulo 19: Cambiar de root (chroot).....	47
Sección 19.1: Requisitos.....	47
Sección 19.2: Cambiar manualmente root de un directorio.....	47
Sección 19.3: Razones para utilizar chroot.....	47
Capítulo 20: Gestores de paquetes.....	49
Sección 20.1: Cómo actualizar paquetes con el gestor de paquetes apt.....	49
Sección 20.2: Cómo instalar un paquete con el gestor de paquetes pacman	49
Sección 20.3: Cómo actualizar paquetes con el gestor de paquetes pacman	49
Sección 20.4: Cómo actualizar paquetes con yum.....	50
Capítulo 21: Compilación del kernel Linux	50
Sección 21.1: Compilación del núcleo Linux en Ubuntu.....	50
Créditos	52

Acerca de

Este libro ha sido traducido por rortegag.com

Si desea descargar el libro original, puede descargarlo desde:

<https://goalkicker.com/LinuxBook/>

Si desea contribuir con una donación, hazlo desde:

<https://www.buymeacoffee.com/GoalKickerBooks>

Por favor, siéntase libre de compartir este PDF con cualquier persona de forma gratuita, la última versión de este libro se puede descargar desde:

<https://goalkicker.com/LinuxBook/>

Este libro Linux® Apuntes para Profesionales está compilado a partir de la [Documentación de Stack Overflow](#), el contenido está escrito por la hermosa gente de Stack Overflow. El contenido del texto está liberado bajo Creative Commons BY-SA, ver los créditos al final de este libro quién contribuyó a los distintos capítulos. Las imágenes pueden ser copyright de sus respectivos propietarios a menos que se especifique lo contrario.

Este es un libro no oficial gratuito creado con fines educativos y no está afiliado con los grupo(s) o empresa(s) oficiales de Linux® ni Stack Overflow. Todas las marcas comerciales y marcas registradas son propiedad de sus respectivos propietarios de la empresa.

No se garantiza que la información presentada en este libro sea correcta ni exacta. Utilícelo bajo su propia responsabilidad.

Envíe sus comentarios y correcciones a web@petercv.com

Capítulo 1: Introducción a GNU/Linux

Sección 1.1: Atajos útiles

Uso del terminal

Los ejemplos de este documento asumen que está utilizando un shell compatible con POSIX (como **bash**, **sh**, **zsh**, **ksh**) shell.

Gran parte de la funcionalidad de GNU/Linux se consigue utilizando el terminal. La mayoría de las distribuciones de Linux incluyen emuladores de terminal que permiten a los usuarios interactuar con una shell desde su entorno de escritorio. Un shell es un intérprete de línea de comandos que ejecuta los comandos introducidos por el usuario. **Bash** (Bourne Again SHell) es un intérprete de comandos común por defecto entre muchas distribuciones de Linux y es el intérprete de comandos por defecto de macOS.

Estos atajos funcionarán si estás usando **Bash** con las combinaciones de teclas de *emacs* (configuradas por defecto):

Abrir terminal

- `Ctrl + Alt + T` o `Super + T`

Movimiento del cursor

- `Ctrl + A` Ir al principio de la línea en la que está escribiendo.
- `Ctrl + E` Ir al final de la línea en la que estás escribiendo.
- `Ctrl + XX` Desplazarse entre el principio de la línea y la posición actual del cursor.
- `Alt + F` Avanza el cursor una palabra en la línea actual.
- `Alt + B` Desplazar el cursor una palabra hacia atrás en la línea actual.
- `Ctrl + F` Avanza el cursor un carácter en la línea actual.
- `Ctrl + B` Desplazar el cursor un carácter hacia atrás en la línea actual.

Manipulación de textos

- `Ctrl + U` Corta la línea desde la posición actual hasta el principio de la línea, añadiéndola al portapapeles. Si está al final de la línea, corta toda la línea.
- `Ctrl + K` Corta la línea desde la posición actual hasta el final de la línea, añadiéndola al portapapeles. Si se encuentra al principio de la línea, corta toda la línea.
- `Ctrl + W` Borrar la palabra situada antes del cursor, añadiéndola al portapapeles.
- `Ctrl + Y` Pega lo último del portapapeles que hayas cortado recientemente (deshace el último borrado en la posición **actual** del cursor).
- `Alt + T` Intercambiar las dos últimas palabras antes del cursor.
- `Alt + L` Hacer minúsculas desde el cursor hasta el final de la palabra.
- `Alt + U` Hacer mayúsculas desde el cursor hasta el final de la palabra.
- `Alt + C` Mayúsculas hasta el final de la palabra que empieza en el cursor (toda la palabra si el cursor está al principio de la palabra).
- `Alt + D` Borrar hasta el final de la palabra empezando por el cursor (toda la palabra si el cursor está al principio de la palabra).
- `Alt + .` Imprime la última palabra escrita en el comando anterior.
- `Ctrl + T` Intercambia los dos últimos caracteres antes del cursor.

Acceso al historial

- `Ctrl + R` Permite buscar entre los comandos utilizados anteriormente.
- `Ctrl + G` Salir del modo de búsqueda en el historial sin ejecutar ningún comando.

- **Ctrl + J** Le permite copiar el comando coincidente actual a la línea de comandos sin ejecutarlo, lo que le permite realizar modificaciones antes de ejecutar el comando.
- **Alt + R** Revierte los cambios de un comando que hayas sacado de tu historial, si lo has editado.
- **Ctrl + P** Muestra el último comando ejecutado, es decir, retrocede en el historial de comandos (similar a la flecha hacia arriba).
- **Ctrl + N** Muestra el siguiente comando ejecutado, es decir, avanza por el historial de comandos (similar a la flecha abajo).

Control de terminales

- **Ctrl + L** Borra la pantalla, de forma similar al comando clear.
- **Ctrl + S** Detener toda la salida a la pantalla. Esto es útil cuando se ejecutan comandos con mucha salida larga. Pero esto no detiene el comando en ejecución.
- **Ctrl + Q** Reanudar la salida a la pantalla después de detenerla con Ctrl+S.
- **Ctrl + C** Finalizar el proceso en ejecución y volver al prompt.
- **Ctrl + D** Salir de la sesión de shell actual, de forma similar al comando exit o logout. En algunos comandos, actúa como señal de fin de archivo para indicar que se ha alcanzado el final de un archivo.
- **Ctrl + Z** Suspende (pausa) el proceso que se está ejecutando en primer plano, lo que devuelve el prompt del shell. A continuación, puede utilizar el comando bg para que ese proceso se ejecute en segundo plano. Para volver a poner ese proceso en primer plano, utilice el comando fg. Para ver todos los procesos en segundo plano, utilice el comando **jobs**.
- **Tabulación** Autocompletar nombres de archivos y directorios.
- **Tabulación Tabulación** Muestra todas las posibilidades, cuando los caracteres tecleados no coinciden unívocamente con un nombre de archivo o directorio.

Caracteres especiales

- **Ctrl + H** Igual que Retroceso.
- **Ctrl + J** Igual que Retorno (históricamente Line Feed).
- **Ctrl + M** Igual que Retorno (históricamente Retorno de carro).
- **Ctrl + I** Igual que Tabulación.
- **Ctrl + G** Carácter Campana.
- **Ctrl + @** Carácter nulo.
- **Esc** Tecla muerta equivalente al modificador Alt.

Cerrar Terminal

- **Ctrl + Mayús + W** Para cerrar la pestaña terminal.
- **Ctrl + Mayús + Q** Para cerrar todo el terminal.

Alternativamente, puedes cambiar a las combinaciones de teclas de vi en **bash** usando **set -o vi**. Usa **set -o emacs** para volver a las combinaciones de teclas de emacs.

Sección 1.2: Comandos de gestión de archivos

Linux utiliza algunas convenciones para los directorios presentes y padres. Esto puede ser un poco confuso para los principiantes.

Siempre que estés en un terminal en Linux, estarás en lo que se llama el *directorio de trabajo actual*. A menudo su símbolo del sistema mostrará el directorio de trabajo completo, o sólo la última parte de ese directorio. Su prompt podría verse como uno de los siguientes:

```
user@host ~/somedir $
user@host somedir $
user@host /home/user/somedir $
```

que dice que tu directorio de trabajo actual es **/home/user/somedir**.

En Linux **..** representa el directorio padre y **.** representa el directorio actual.

Por lo tanto, si el directorio actual es `/home/user/somedir`, entonces `cd ../somedir` no cambiará el directorio de trabajo.

En la tabla siguiente se enumeran algunos de los comandos de gestión de archivos más utilizados.

Navegación por el directorio

Comando	Utilidad
<code>pwd</code>	Obtener la ruta completa del directorio de trabajo actual.
<code>cd -</code>	Navigate hasta el último directorio en el que estaba trabajando.
<code>cd ~</code> o sólo <code>cd</code>	Navega al directorio personal del usuario actual.
<code>cd ..</code>	Ir al directorio padre del directorio actual (tenga en cuenta el espacio entre <code>cd</code> y <code>..</code>)

Listado de archivos dentro de un directorio

Comando	Utilidad
<code>ls -l</code>	Lista los archivos y directorios en el directorio actual en formato largo (tabla) (Se recomienda usar <code>-l</code> con <code>ls</code> para una mejor legibilidad).
<code>ls -ld dir-name</code>	Muestra información sobre el directorio <code>dir-name</code> en lugar de su contenido.
<code>ls -a</code>	Lista todos los archivos, incluidos los ocultos (los nombres de archivo que empiezan por <code>.</code> son archivos ocultos en Linux).
<code>ls -F</code>	Añade un símbolo al final del nombre de un archivo para indicar su tipo (<code>*</code> significa ejecutable, <code>/</code> significa directorio, <code>@</code> significa enlace simbólico, <code>=</code> significa socket, <code> </code> significa tubería con nombre, <code>></code> significa puerta).
<code>ls -lt</code>	Lista los archivos ordenados por la última vez que se modificaron, mostrando los archivos modificados más recientemente en la parte superior (recuerde que la opción <code>-l</code> proporciona el formato largo, que es más legible).
<code>ls -lh</code>	Enumera los tamaños de los archivos en un formato legible.
<code>ls -lR</code>	Muestra todos los subdirectorios de forma recursiva.
<code>tree</code>	Generará una representación en árbol del sistema de archivos a partir del directorio actual.

Creación, copia y eliminación de archivos/directorios

Comando	Utilidad
<code>cp -p source destination</code>	Copiará el archivo <code>source</code> al <i>destino</i> . <code>-p</code> significa preservación. Conserva los atributos originales del archivo durante la copia, como el propietario del archivo, la fecha y hora, el grupo, los permisos, etc.
<code>cp -R source_dir destination_dir</code>	Copiará el directorio de origen en el destino especificado de forma recursiva.
<code>mv file1 file2</code>	En Linux no existe el comando renombrar como tal. Por lo tanto <code>mv</code> mueve/renombrará el <code>file1</code> al <code>file2</code> .
<code>rm -i filename</code>	Le pregunta antes de cada eliminación de archivos para confirmar. SI ERES UN NUEVO USUARIO DE LINUX, SIEMPRE DEBE UTILIZAR <code>rm -i</code> . Puede especificar múltiples archivos.
<code>rm -R dir-name</code>	Eliminará el directorio <code>dir-name</code> recursivamente.
<code>rm -rf dir-name</code>	Eliminará el directorio <code>dir-name</code> recursivamente, ignorando archivos no existentes y nunca preguntará nada. ¡TENGA CUIDADO AL USAR ESTE COMANDO! Puede especificar varios directorios.
<code>rmdir dir-name</code>	Eliminará el directorio <code>dir-name</code> , si está vacío. Este comando sólo puede eliminar directorios vacíos.
<code>mkdir dir-name</code>	Crea un directorio <code>dir-name</code> .
<code>mkdir -p dir-name/dir-name</code>	Cree una jerarquía de directorios. Cree los directorios padres que sean necesarios, si no existen. Puede especificar varios directorios.

touch filename

Crea un archivo **filename**, si no existe, de lo contrario cambia la marca de tiempo del archivo a la hora actual.

Permisos y grupos de archivos/directorios

Comandos

chmod <specification> filename

chmod -R <specification> dir- name

chmod go=+r myfile

chmod a +rwx myfile

chmod go -r myfile

chown owner1 filename

chgrp grp_owner filename

chgrp -R grp_owner dir-name

Utilidad

Cambia los permisos del archivo. Especificaciones = **u** usuario, **g** grupo, **o** otro, **+** añadir permiso, **-** eliminar, **r** leer, **w** escribir, **x** ejecutar.

Cambia los permisos de un directorio de forma recursiva. Para cambiar los permisos de un directorio y de todo lo que contiene, utilice este comando.

Añade permiso de lectura para el propietario y el grupo. Permitir a todos los usuarios leer, escribir o ejecutar **myfile**. Quitar el permiso de lectura al grupo y a los demás.

Cambiar la propiedad de un archivo al usuario **owner1**.

Cambia la propiedad del grupo primario del archivo **filename** al grupo **grp_owner**.

Cambia la propiedad del grupo primario del directorio **dir-name** al grupo **grp_owner** recursivamente. Para cambiar la propiedad de grupo de un directorio y todo lo que hay dentro de ese directorio, utiliza este comando.

Sección 1.3: Hola Mundo

Escriba el siguiente código en su terminal y pulse **Enter**:

echo "Hello World"

Esto producirá el siguiente resultado:

Hello World

Sección 1.4: Utilidades básicas de Linux

Linux tiene un comando para casi cualquier tarea y la mayoría de ellos son intuitivos y fáciles de interpretar.

Obtener ayuda en Linux

Comando

man <name>

man <section> <name>

man -k <editor>

man -K <keyword>

apropos <editor>

help

help <name>

info <name>

dpkg -l

Usabilidad

Lee la página del manual de <name>.

Leer la página del manual de <name>, relacionada con la sección dada.

Muestra todos los programas cuyas páginas de manual contienen la palabra clave <editor>.

Muestra todas las páginas de manual que contienen <keyword>.

Muestra todas las aplicaciones cuya descripción de una línea coincide con la palabra editor. Si **no recuerda** el nombre de la aplicación, utilice este comando.

En el shell Bash, esto mostrará la lista de todos los comandos bash disponibles.

En el shell Bash, esto mostrará la información sobre el comando bash <name>.

Ver toda la información sobre <name>.

Muestra una lista de todos los paquetes instalados en un sistema basado en Debian.

<code>dpkg -L packageName</code>	Muestra los archivos instalados y los detalles de la ruta de un paquete determinado en Debian.
<code>dpkg -l grep -i <edit></code>	Devuelve todos los paquetes .deb instalados con <edit> independientemente de los casos.
<code>less /var/lib/dpkg/available</code>	Devuelve descripciones de todos los paquetes disponibles.
<code>whatis vim</code>	Enumera una descripción de una línea de vim.
<code><command-name> --help</code>	Muestra información de uso sobre <nombre-de-la-herramienta>. A veces el <code>command -h</code> también funciona, pero no para todos los comandos.

Identificación de usuarios y quién es quién en el mundo Linux

Comando	Usabilidad
<code>hostname</code>	Muestra el nombre de host del sistema.
<code>hostname -f</code>	Muestra el nombre de dominio completo (FQDN) del sistema.
<code>passwd</code>	Cambiar la contraseña del usuario actual.
<code>whoami</code>	Nombre de usuario de los usuarios registrados en el terminal.
<code>who</code>	Lista de todos los usuarios actualmente conectados como usuario.
<code>w</code>	Muestra el estado actual del sistema, la hora, la duración, la lista de usuarios conectados actualmente al sistema y otra información sobre los usuarios.
<code>last</code>	Quién ha utilizado recientemente el sistema.
<code>last root</code>	Cuándo fue la última vez que root se conectó como usuario.
<code>lastb</code>	Muestra todos los intentos fallidos de inicio de sesión en el sistema.
<code>chmod</code>	Cambio de permisos - lectura, escritura, ejecución de un archivo o directorio.

Información relacionada con el proceso

Comando	Usabilidad
<code>top</code>	Lista todos los procesos ordenados por su uso actual de recursos del sistema. Muestra una visualización continuamente actualizada de los procesos (Por defecto 3 segundos). Utilice la tecla q para salir de la parte superior.
<code>ps</code>	Lista de procesos que se están ejecutando en la sesión de shell actual.
<code>ps -u root</code>	Lista de todos los procesos y comandos que root está ejecutando.
<code>ps aux</code>	Lista todos los procesos de todos los usuarios en el sistema actual.

Sección 1.5: Búsqueda de archivos por patrones en el nombre/contenido

Una tarea común de alguien que utiliza la línea de comandos (shell) de Linux es buscar archivos/directorios con un nombre determinado o que contengan cierto texto. Hay 2 comandos con los que deberías familiarizarte para conseguirlo:

Buscar archivos por nombre

```
find /var/www -name '*.css'
```

Esto imprimirá la ruta completa/nombre de todos los archivos bajo /var/www que terminen en .css. Ejemplo de salida:

```
/var/www/html/text-cursor.css
/var/www/html/style.css
```

Para más información:

```
man find
```

Buscar archivos que contengan texto

```
grep font /var/www/html/style.css
```

Esto imprimirá todas las líneas que contengan la fuente del patrón en el archivo especificado. Ejemplo de salida:

```
font-weight: bold;
font-family: monospace;
```

Otro ejemplo:

```
grep font /var/www/html/
```

Esto no funciona como esperabas. Lo consigues:

```
grep: /var/www/html/: Is a directory
```

Necesitas hacer **grep** recursivo para que funcione, usando la opción **-R**:

```
grep -R font /var/www/html/
```

¡Eh, bien! Echa un vistazo a la salida de este:

```
/var/www/html/admin/index.php: echo '<font color=red><b>Error: no dice</b></font><br/>';
/var/www/html/admin/index.php: echo '<font color=red><b>Error: try again</b></font><br/>';
/var/www/html/style.css: font-weight: bold;
/var/www/html/style.css: font-family: monospace;
```

Fíjese que cuando **grep** está comparando múltiples ficheros, prefija las líneas comparadas con los nombres de los ficheros. Si lo desea, puede utilizar la opción **-h** para eliminarlo.

Para más información:

```
man grep
```

Sección 1.6: Manipulación de archivos

Los archivos y directorios (otro nombre para las carpetas) son el corazón de Linux, por lo que poder crearlos, verlos, moverlos y borrarlos desde la línea de comandos es muy importante y bastante potente. Estos comandos de manipulación de archivos le permiten realizar las mismas tareas que realizaría un explorador gráfico de archivos.

Crea un archivo de texto vacío llamado **myFile**:

```
touch myFile
```

Cambia el nombre de **myFile** a **myFirstFile**:

```
mv myFile myFirstFile
```

Ver el contenido de un archivo:

```
cat myFirstFile
```

Visualiza el contenido de un archivo con el paginador (una pantalla a la vez):

```
less myFirstFile
```

Ver las primeras líneas de un archivo:

```
head myFirstFile
```

Visualiza las últimas líneas de un archivo:

```
tail myFirstFile
```

Editar un archivo:

```
vi myFirstFile
```

Vea qué archivos hay en su directorio de trabajo actual:

```
ls
```

Crea un directorio vacío llamado `myFirstDirectory`:

```
mkdir myFirstDirectory
```

Crear directorio multiruta: (crea dos directorios, `src` y `myFirstDirectory`)

```
mkdir -p src/myFirstDirectory
```

Mueva el archivo al directorio:

```
mv myFirstFile myFirstDirectory/
```

También puede cambiar el nombre del archivo:

```
user@linux-computer:~$ mv myFirstFile secondFileName
```

Cambia el directorio de trabajo actual a `myFirstDirectory`:

```
cd myFirstDirectory
```

Borrar un archivo:

```
rm myFirstFile
```

Mover al directorio padre (que se representa como `..`):

```
cd ..
```

Borrar un directorio vacío:

```
rmdir myFirstDirectory
```

Eliminar un directorio no vacío (es decir, que contenga archivos y/u otros directorios):

```
rm -rf myFirstDirectory
```

Ten en cuenta que cuando borres directorios, debes borrar `./` y no `/`, ya que se borraría todo el sistema de archivos.

Sección 1.7: Detalles del archivo/directorio

El comando `ls` dispone de varias opciones que pueden utilizarse conjuntamente para mostrar más información.

Detalles/Derechos

La opción `l` muestra los permisos del archivo, el tamaño y la fecha de la última modificación. Así que si el directorio raíz contiene un `dir` llamado `test` y un archivo `someFile` el comando:

```
user@linux-computer:~$ ls -l
```

Saldría algo como

```
-rw-r--r-- 1 user users 70 Jul 22 13:36 someFile.txt
drwxrwxrwx 2 user users 4096 Jul 21 07:18 test
```

Los permisos tienen el formato `drwxrwxrwx`. El primer carácter representa el tipo de archivo `d` si es un directorio `-` de lo contrario. Los tres siguientes `rw` son los permisos que el usuario tiene sobre el archivo, los tres siguientes son los permisos que el grupo tiene sobre el archivo, y los tres últimos son los permisos que todos los demás tienen sobre el archivo.

La `r` de `rw` representa si un archivo puede ser leído, la `w` representa si el archivo puede ser modificado, y la `x` representa si el archivo puede ser ejecutado. Si no se concede ningún permiso, aparecerá un `-` en lugar de `r`, `w` o `x`.

Así que desde arriba el usuario puede leer y modificar `someFile.txt` pero el grupo sólo tiene derechos de sólo lectura.

Para cambiar los derechos puedes usar el comando `chmod ### fileName` si tienes derechos sudo. `r` se representa con un valor de 4, `w` se representa con 2, y `x` se representa con un 1. Así que si sólo quieres poder modificar el contenido al directorio de `test`.

```
Owner  rwx = 4+2+1 = 7
Group  r-x = 4+0+1 = 5
Other  r-x = 4+0+1 = 5
```

Así que todo el comando es

```
chmod 755 test
```

Ahora haciendo un `ls -l` mostraría algo como

```
drwxr-xr-x 2 user users 4096 Jul 21 07:20 test
```

Tamaño legible

Utilizada junto con la opción `l`, la opción `h` muestra tamaños de archivo legibles. Ejecutar

```
user@linux-computer:~$ ls -lh
```

Saldría:

```
total 4166
-rw-r--r-- 1 user users 70 Jul 22 13:36 someFile.txt
drwxrwxrwx 2 user users 4.0K Jul 21 07:18 test
```

Oculto

Para ver los archivos ocultos, utilice la opción `a`. Por ejemplo

```
user@linux-computer:~$ ls -a
```

Lista posible

```
.profile
someFile.txt
test
```

Tamaño total del directorio

Para ver el tamaño del directorio actual utilice la opción `s` (también se puede utilizar la opción `h` para que el tamaño sea más legible).

```
user@linux-computer:~$ ls -s
```

Salidas

```
total 4166 someFile.txt
test
```

Vista recursiva

Supongamos que el directorio de `test` tiene un archivo `anotherFile` y usted quiere verlo desde la carpeta raíz, puede usar la opción `R` que listará el árbol recursivo.

```
user@linux-computer:~$ ls -R
```

Salidas

```
.:
someFile.txt test
./test:
anotherFile
```

Capítulo 2: Detección del nombre y la versión de la distribución Linux

Sección 2.1: Detecte en qué distribución basada en Debian está trabajando

Basta con ejecutar `lsb_release -a`.

En Debian:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description: Debian GNU/Linux testing (stretch)
Release: testing Codename: stretch
```

En Ubuntu:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 14.04.4 LTS
Release: 14.04 Codename: trusty
```

En caso de que no tenga `lsb_release` instalado, puede intentar adivinarlo, por ejemplo, hay un archivo `/etc/issue` que a menudo contiene el nombre de la distribución. Por ejemplo, en Ubuntu:

```
$ cat /etc/issue
Ubuntu 12.04.5 LTS \n \l
```

No utilice el fichero `/etc/debian_version` porque su contenido no coincide con el nombre de la distribución.

Tenga en cuenta que esto también funcionará en distribuciones que no sean de la familia Debian, como Fedora, RHEL u openSUSE, pero es posible que `lsb_release` no esté instalado.

Sección 2.2: Detectar qué distribución basada en systemd está utilizando

Este método funcionará en versiones modernas de Arch, CentOS, CoreOS, Debian, Fedora, Mageia, openSUSE, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Ubuntu y otros. Esta amplia aplicabilidad lo hace ideal como primera aproximación, con la posibilidad de recurrir a otros métodos si necesita identificar también sistemas más antiguos.

Mira en `/etc/os-release`. En específico, mire las variables `NAME`, `VERSION`, `ID`, `VERSION_ID` y `PRETTY_NAME`. En Fedora, este archivo podría tener este aspecto:

```
NAME=Fedora
VERSION="24 (Workstation Edition)"
ID=fedora
VERSION_ID=24
PRETTY_NAME="Fedora 24 (Workstation Edition)"
ANSI_COLOR="0;34"
CPE_NAME="cpe:/o:fedoraproject:fedora:24"
HOME_URL="https://fedoraproject.org/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"
REDHAT_BUGZILLA_PRODUCT="Fedora"
REDHAT_BUGZILLA_PRODUCT_VERSION=24
REDHAT_SUPPORT_PRODUCT="Fedora"
REDHAT_SUPPORT_PRODUCT_VERSION=24
PRIVACY_POLICY_URL=https://fedoraproject.org/wiki/Legal:PrivacyPolicy
VARIANT="Workstation Edition"
VARIANT_ID=workstation
```

En CentOS, este archivo puede tener este aspecto:

```
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"
CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"
```

Este archivo está [documentado en el sitio web de freedesktop](#); en principio, no es específico de systemd - pero existirá en todas las distribuciones basadas en systemd.

Desde el intérprete de comandos bash, se puede acceder al archivo `/etc/os-release` y utilizar directamente las distintas variables, como se indica a continuación:

```
$ ( source /etc/os-release && echo "$PRETTY_NAME" )
Fedora 24 (Workstation Edition)
```

Sección 2.3: Detectar en qué distribución RHEL / CentOS / Fedora está trabajando

Mira el contenido de `/etc/redhat-release`

```
cat /etc/redhat-release
```

Aquí está la salida de una máquina Fedora 24: `Fedora release 24 (Twenty Four)`

Como se menciona en la respuesta basada en debian, también puede utilizar el comando `lsb_release -a`, que muestra esto desde una máquina Fedora 24:

```
LSB Version: :core-4.1-amd64:core-4.1-noarch:cxx-4.1-amd64:cxx-4.1-noarch:desktop-4.1-
amd64:desktop-4.1-noarch:languages-4.1-amd64:languages-4.1-noarch:printing-4.1-amd64:printing-
4.1-noarch
Distributor ID: Fedora
Description: Fedora release 24 (Twenty Four)
Release: 24
Codename: TwentyFour
```

Sección 2.4: Uname - Imprime información sobre el sistema actual

Uname es el nombre abreviado de **unix name**. Sólo tienes que escribir `uname` en la consola para obtener información sobre tu sistema operativo.

```
uname [OPTION]
```

Si no se especifica ninguna OPCIÓN, `uname` asume la opción `-s`.

`-a` o `--all` - Imprime toda la información, omitiendo `-p` y `-i` si la información es desconocida.

Ejemplo:

```
> uname -a
SunOS hope 5.7 Generic_106541-08 sun4m sparc SUNW,SPARCstation-10
```

Todas las opciones:

-s, --kernel-name	Imprime el nombre del núcleo.
-n, --nodename	Imprime el nombre de host del nodo de red.
-r, --kernel-release	Imprime la versión del kernel.
-v, --kernel-version	Imprime la versión del kernel.
-m, --machine	Imprime el nombre del hardware de la máquina.
-p, --processor	Imprime el tipo de procesador, o "unknown".
-i, --hardware-platform	Imprime la plataforma de hardware, o "unknown".
-o, --operating-system	Imprime el sistema operativo.
--help	Mostrar un mensaje de ayuda y salir.
--version	Mostrar información sobre la versión y salir.

Sección 2.5: Detectar información básica sobre su distribución

Simplemente ejecute `uname -a`.

En Arch:

```
$ uname -a
Linux nokia 4.6.4-1-ARCH #1 SMP PREEMPT Mon Jul 11 19:12:32 CEST 2016 x86_64
GNU/Linux enter code here
```

Sección 2.6: Utilizar GNU coreutils

Así que GNU coreutils debería estar disponible en todos los sistemas basados en Linux (por favor, corrígeme si me equivoco).

Si no sabe qué sistema está utilizando, es posible que no pueda saltar directamente a uno de los ejemplos anteriores, por lo que éste puede ser su primer puerto de escala.

```
$ uname -a
```

En mi sistema esto me da lo siguiente...

```
Linux Scibearspace 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt25-2+deb8u3 (2016-07-02) x86_64
GNU/Linux
```

Aquí puede ver lo siguiente:

Scibearspace: el nombre de mi pc

- Scibearspace: el nombre de mi pc
- 3.16.0-4-amd64: el núcleo y la arquitectura
- SMP Debian 3.16.7-CKT25-2+deb8u3: me dice que estoy ejecutando debian con el kernel 3.16
- Finalmente, la última parte que estoy ejecutando debian 8 (actualización 3).

Me gustaría que otros añadieran los resultados de los sistemas RHEL y SuSe.

Sección 2.7: Encuentre el nombre y el número de versión de su sistema operativo Linux (tanto debian como rpm)

La mayoría de las distribuciones linux almacenan su información de versión en el archivo `/etc/lsb-release` (debian) o `/etc/redhat-release` (basado en RPM). Usando el comando genérico de abajo debería pasar la mayoría de los derivados de Debian y RPM como Linux Mint y Cent-Os.

Ejemplo en máquina Ubuntu:

```
cat /etc/*release
```



```
DISTRIB_ID=Ubuntu  
DISTRIB_RELEASE=14.04  
DISTRIB_CODENAME=trusty  
DISTRIB_DESCRIPTION="Ubuntu 14.04 LTS"
```

Capítulo 3: Obtener información sobre un núcleo Linux en ejecución

Sección 3.1: Obtener detalles del núcleo de Linux

Podemos usar el comando `uname` con varias opciones para obtener detalles completos del kernel en ejecución.

`uname -a`

```
Linux df1-ws-5084 4.4.0-64-generic #85-Ubuntu SMP Mon Feb 20 11:50:30 UTC 2017 x86_64 x86_64  
x86_64 GNU/Linux
```

Según la página de manual aquí hay algunas opciones más

Uso: `uname [OPTION]...`

Imprime cierta información del sistema. Sin `OPTION`, igual que `-s`.

```
-a, --all      print all information, in the following order, except omit -p and -i if unknown:  
-s, --kernel-name      print the kernel name  
-n, --nodename  print the network node hostname  
-r, --kernel-release print the kernel release  
-v, --kernel-version print the kernel version  
-m, --machine   print the machine hardware name  
-p, --processor  print the processor type (non-portable)  
-i, --hardware-platform  print the hardware platform (non-portable)  
-o, --operating-system  print the operating system  
--help         display this help and exit  
--version      output version information and exit
```

Capítulo 4: Shell

El shell ejecuta un programa en respuesta a su prompt. Cuando das una orden, el shell busca el programa y lo ejecuta. Por ejemplo, cuando das el comando `ls`, el shell busca la utilidad/programa llamado `ls`, y luego lo ejecuta en el shell. Los argumentos y las opciones que proporcionas con las utilidades pueden influir en el resultado que obtengas. El shell también se conoce como CLI, o interfaz de línea de comandos.

Sección 4.1: Cambiar el shell por defecto

La mayoría de las distribuciones modernas vienen con BASH (**B**ourne **A**gain **S**hell) preinstalado y configurado como shell por defecto.

El comando (en realidad un binario ejecutable, un ELF) responsable de cambiar de shell en Linux es `chsh` (**ch**ange **sh**ell).

En primer lugar, podemos comprobar qué shells están ya instalados y configurados en nuestra máquina utilizando el comando `chsh -l`, que mostrará un resultado similar a este:

```
[user@localhost ~]$ chsh -l
/bin/sh
/bin/bash
/sbin/nologin
/usr/bin/sh
/usr/bin/bash
/usr/sbin/nologin
/usr/bin/fish
```

En algunas distribuciones de Linux, `chsh -l` no es válido. En este caso, la lista de todos los shells disponibles se puede encontrar en el archivo `/etc/shells`. Puede mostrar el contenido del archivo con `cat`:

```
[user@localhost ~]$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/sbin/nologin
/usr/bin/sh
/usr/bin/bash
/usr/sbin/nologin
/usr/bin/fish
```

Ahora podemos elegir nuestro nuevo shell por defecto, por ejemplo `fish`, y configurarlo usando `chsh -s`,

```
[user@localhost ~]$ chsh -s /usr/bin/fish
Changing shell for user.
Password:
Shell changed.
```

Ahora todo lo que queda por hacer es preformar un ciclo cerrar-iniciar sesión, y disfrutar de nuestro nuevo shell por defecto.

Si deseas cambiar el shell por defecto para un usuario diferente, y tienes privilegios administrativos en la máquina, podrás lograrlo usando `chsh` como `root`. Así que asumiendo que queremos cambiar el shell por defecto del `user_2` a `fish`, usaremos el mismo comando que antes, pero añadiendo el nombre de usuario del otro usuario, `chsh -s /usr/bin/fish user_2`.

Para comprobar cuál es el shell por defecto actual, podemos ver la variable de entorno `$SHELL`, que apunta a la ruta de nuestro shell por defecto, por lo que después de nuestro cambio, esperaríamos obtener un resultado similar a este,

```
~$ echo $SHELL
/usr/bin/fish
```

opciones **chsh**:

-s shell

Establece shell como shell de inicio de sesión.

-l, **--list-shells**

Imprime la lista de shells listados en `/etc/shells` y sale.

-h, **--help**

Imprime un mensaje de uso y sale.

-v, **--version**

Imprimir la información de la versión y salir.

Sección 4.2: Utilidades básicas del shell

Personalizar el prompt de Shell

El símbolo del sistema por defecto se puede cambiar para que tenga un aspecto diferente y corto. En caso de que el directorio actual sea largo, el símbolo del sistema por defecto se vuelve demasiado grande. El uso de `PS1` es útil en estos casos. Un comando corto y personalizado bonito y elegante. En la siguiente tabla se ha utilizado `PS1` con un número de argumentos para mostrar diferentes formas de prompt de shell. El prompt de comandos por defecto se parece a esto: `user@host ~ $` en mi caso se parece a esto: `bruce@gotham ~ $` Se puede cambiar según la siguiente tabla:

Comando	Utilidad
<code>PS1='\w \$'</code>	<code>~ \$</code> prompt del shell como nombre de directorio. En este caso el directorio raíz es Root.
<code>PS1='\h \$'</code>	<code>gotham \$</code> prompt del shell como nombre de host
<code>PS1='\u \$'</code>	<code>bruce \$</code> prompt del shell como nombre de usuario
<code>PS1='\t \$'</code>	<code>22:37:31 \$</code> prompt del shell en formato de 24 horas
<code>PS1='@ \$'</code>	<code>10:37 PM</code> prompt del shell en formato de 12 horas
<code>PS1='! \$'</code>	<code>732</code> mostrará el número de historia del comando en lugar del prompt del shell
<code>PS1='dude \$'</code>	<code>dude \$</code> mostrará el intérprete de comandos de la forma que desee

Algunos comandos básicos del shell

Comando	Utilidad
<code>Ctrl-k</code>	cortar/matar
<code>Ctrl-y</code>	tirar/pegar
<code>Ctrl-a</code>	llevará el cursor al principio de la línea
<code>Ctrl-e</code>	llevará el cursor al final de la línea
<code>Ctrl-d</code>	borrará el carácter después/en el cursor
<code>Ctrl-l</code>	borrará la pantalla/terminal
<code>Ctrl-u</code>	borrará todo lo que haya entre el prompt y el cursor
<code>Ctrl-_</code>	deshará lo último escrito en la línea de comandos
<code>Ctrl-c</code>	interrumpirá/detendrá el trabajo/proceso que se esté ejecutando en primer plano
<code>Ctrl-r</code>	búsqueda inversa en la historia
<code>~/.bash_history</code>	almacena los últimos 500 comandos/eventos utilizados en el shell
history	mostrará el historial de comandos
history grep <key-word>	mostrará todos los comandos del historial que tengan la palabra clave <code><palabra clave></code> (útil en casos en los que recuerde parte del comando utilizado en el pasado)

Sección 4.3: Cree su propio alias de comando

Si estás cansado de usar comandos largos en bash puedes crear tu propio alias de comando.

La mejor forma de hacerlo es modificar (o crear si no existe) un archivo llamado `.bash_aliases` en tu carpeta home. La sintaxis general es:

```
alias command_alias='actual_command'
```

donde `actual_command` es el comando que está renombrando y `command_alias` es el nuevo nombre que le ha dado. Por ejemplo

```
alias install='sudo apt-get -y install'
```

asigna el nuevo alias de comando `install` al comando real `sudo apt-get -y install`. Esto significa que cuando usas `install` en una terminal, bash lo interpreta como `sudo apt-get -y install`.

Sección 4.4: Localiza un archivo en tu sistema

Usando bash puedes localizar fácilmente un archivo con el comando `locate`. Por ejemplo, digamos que usted está buscando el archivo `mykey.pem`:

```
locate mykey.pem
```

A veces los archivos tienen nombres extraños por ejemplo usted podría tener un archivo como `random7897_mykey_0fidw.pem`. Digamos que estás buscando este archivo, pero sólo recuerdas las partes `mykey` y `pem`. Podrías combinar el comando `locate` con `grep` usando una tubería como esta:

```
locate pem | grep mykey
```

Lo que mostraría todos los resultados que contengan estas dos piezas.

Tenga en cuenta que no todos los sistemas tienen instalada la utilidad `locate`, y muchos de los que la tienen no la han habilitado. `locate` es rápida y eficiente porque escanea periódicamente su sistema y almacena en caché los nombres y ubicaciones de todos los archivos que hay en él, pero si esa recopilación de datos no está habilitada no puede decirle nada. Puede utilizar `updatedb` para iniciar manualmente el escaneo del sistema de archivos para actualizar la información almacenada en caché sobre los archivos de su sistema de archivos.

Si no dispone de una función de `locate`, puede recurrir a la utilidad de `find`:

```
find / -name mykey.pem -print
```

es más, o menos equivalente a `locate mykey.pem` pero tiene que escanear tu(s) sistema(s) de archivos cada vez que lo ejecutas en busca del archivo en cuestión, en lugar de utilizar datos almacenados en caché. Esto es obviamente más lento y menos eficiente, pero más en tiempo real. La utilidad `find` puede hacer mucho más que encontrar ficheros, pero una descripción completa de sus capacidades está más allá del alcance de este ejemplo.

Capítulo 5: Comprobar el espacio en disco

Sección 5.1: Investigar el uso del disco en los directorios

A veces puede ser necesario averiguar qué directorio consume cuánto espacio en disco, especialmente cuando se utiliza `df -h` y se dio cuenta de que su espacio disponible en disco es bajo.

du:

El comando `du` resume el uso de disco del conjunto de ARCHIVOS, recursivamente para los directorios.

A menudo se utiliza con la opción `-sh`:

```
-s, --summarize
display only a total for each argument
-h, --human-readable
print sizes in human readable format (e.g., 1K 234M 2G)
```

Para resumir los usos de disco de los archivos en el directorio actual que utilizamos:

```
du -sh *
```

Ejemplo de salida:

572K	Documents
208M	Downloads
4,0K	Music
724K	Pictures
4,0K	Public
4,0K	Templates
4,0K	Videos

También podemos incluir archivos ocultos con el uso de:

```
du -sh .[!.*]
```

Ejemplo de salida:

```
6,3M      .atom
4,0K      .bash_history
4,0K      .bash_logout
8,0K      .bashrc
350M      .cache
195M      .config
12K       .dbus
4,0K      .dmrc
44K       .gconf
60K       .gem
520K      .gimp-2.8
28K       .gnome
4,0K      .ICEauthority
8,3M      .local
8,0K      .nano
404K      .nv
36K       .pki
4,0K      .profile
8,0K      .ssh
0         .sudo_as_admin_successful
4,0K      .Xauthority
4,0K      .xsession-errors
4,0K      .xsession-errors.old
572K      Documents
208M      Downloads
4,0K      Music
724K      Pictures
4,0K      Public
4,0K      Templates
4,0K      Videos
```

En tercer lugar, puede añadir total a la salida añadiendo la opción, `-c`:

```
du -sch .[!.*]
```

Resultado:

```
...
4,0K      Templates
4,0K      Videos
769M      total
```

Lo más importante es utilizar el comando `du` correctamente en el directorio raíz para averiguar qué aplicación/servicio o usuario está consumiendo mucho espacio en disco. Por ejemplo, en el caso de un nivel ridículamente bajo de disponibilidad de espacio en disco para un servidor web y de correo, la razón podría ser un ataque de spam a su servicio de correo y se puede diagnosticar con sólo utilizar el comando `du`.

Investiga el uso del disco en el directorio raíz:

```
sudo du -sch /.[!.*]
```

Ejemplo de salida:

```
16K      /.VolumeIcon.icns
24K      /.VolumeIcon.png
13M      /bin
57M      /boot
4,0K     /cdrom
620K     /dev
13M      /etc
779M     /home
0        /initrd.img
406M     /lib
3,9M     /lib32
4,0K     /lib64
16K      /lost+found
4,0K     /media
4,0K     /mnt
367M     /opt
du: cannot access '/proc/18221/task/18221/fd/4': No such file or directory
du: cannot access '/proc/18221/task/18221/fdinfo/4': No such file or directory
du: cannot access '/proc/18221/fd/4': No such file or directory
du: cannot access '/proc/18221/fdinfo/4': No such file or directory
0        /proc
20K      /root
du: cannot access '/run/user/1000/gvfs': Permission denied
9,4M     /run
13M      /sbin
4,0K     /srv
0        /sys
72K      /tmp
3,5G     /usr
639M     /var
0        /vmlinuz
5,8G     total
```

Por último, el mejor método se forma cuando se añade un valor umbral de tamaño para los directorios para ignorar los pequeños. Este comando sólo mostrará las carpetas con más de 1 GB de tamaño que se encuentran bajo el directorio raíz hasta la rama más lejana de todo el árbol de directorios en su sistema de archivos:

```
sudo du --threshold=1G -ch /.[!..]* /*
```

Ejemplo de salida:

```
1,4G     /usr/lib
1,8G     /usr/share
3,5G     /usr
5,8G     total
```

Sección 5.2: Comprobar el espacio en disco

Es bastante común querer comprobar el estado de las distintas particiones/unidades de disco de tu servidor/ordenador para ver cómo de llenas están. El siguiente comando es el que querrás ejecutar:

```
df -h
```

Esto producirá una salida similar a la siguiente:

```
[root@mail ~]# df -h
Filesystem              Size      Used    Avail   Use%    Mounted on
/dev/mapper/VolGroup-lv_root 19G      1.6G     16G      9%      /
tmpfs                   245M      0         245M     0%      /dev/shm
/dev/sda1                485M     47M      413M     11%     /boot
```

En este ejemplo básico, podemos ver que la partición / sólo tiene un 9% de uso.

Para un ejemplo más complejo que también cubre el uso de `df` para ver varios puntos de montaje, vea a continuación:

```
[root@mail ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/VG-root	1.9T	1.7T	89G	95%	/
/dev/mapper/VG-var	431G	145G	264G	36%	/var
devtmpfs	7.8G	204K	7.8G	1%	/dev
tmpfs	7.8G	4.0K	7.8G	1%	/dev/shm
/dev/md1	495M	126M	344M	27%	/boot
ku.example.com:9421	2.5T	487G	2.0T	20%	/mnt/ test
tmpfs	500M	86M	415M	18%	/var/ngx_pagespeed_cache

En este ejemplo, tenemos una partición `/` que está llena en un `95%` junto con una partición adicional `/var` que sólo está llena en un `36%`.

Tiene un montaje de red externo de `2T` que está montado en `/mnt/test` y un montaje ramdisk/tmpfs de `500M` montado en `/var/ngx_pagespeed_cache`.

Capítulo 6: Obtener información del sistema

Colección de comandos para obtener información relacionada con el sistema.

Sección 6.1: Estadísticas sobre CPU, Memoria, Red y Disco (operaciones de E/S)

Para obtener estadísticas generales sobre los principales componentes de la familia Linux de comandos **stat** son extremadamente útiles.

CPU

Para obtener estadísticas relacionadas con los procesadores puede utilizar el comando **mpstat**, pero con algunas opciones proporcionará una mejor visibilidad:

```
$ mpstat 2 10
```

Memoria

Todos conocemos el comando **free** para mostrar la cantidad de RAM (restante) pero para ver todas las estadísticas incluyendo las operaciones de E/S:

```
$ vmstat 2 10
```

Disco

Para obtener información general sobre las operaciones de tu disco en tiempo real puedes utilizar **iostat**.

```
$ iostat -kx 2
```

Red

Para poder ver lo que ocurre con tus servicios de red puedes utilizar **netstat**

```
$ netstat -ntlp # open TCP sockets
$ netstat -nulp # open UDP sockets
$ netstat -nxlp # open Unix sockets
```

Pero puedes encontrar monitorización útil para ver el tráfico de la red en tiempo real:

```
$ sudo iftop
```

Opcional

Para generar estadísticas en tiempo real relacionadas con las operaciones de E/S en todos los componentes puede utilizar **dsstat**. Esa herramienta que es un reemplazo versátil para **vmstat**, **iostat** y **ifstat**.

Sección 6.2: Uso de herramientas como lscpu y lshw

Mediante el uso de herramientas como **lscpu** como **lscpu** es una manera fácil de obtener información de la CPU.

```
$ lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 4
On-line CPU(s) list: 0-3
Thread(s) per core: 1
Core(s) per socket: 4
Socket(s): 1
NUMA node(s): 1
```

```
Vendor ID: GenuineIntel
CPU family: 6
Model: 23
Stepping: 10
CPU MHz: 1998.000
BogoMIPS: 5303.14
Virtualization: VT-x
L1d cache: 32K
L1i cache: 32K
L2 cache: 2048K
NUMA node0 CPU(s): 0-3
```

Utilizando la herramienta **lshw**

```
$ lshw | grep cpu
df1-ws-5084
description: Computer
width: 64 bits
capabilities: vsyscall32
*-core
description: Motherboard
physical id: 0
*-memory
description: System memory
physical id: 0
size: 5881MiB
*-cpu
product: Intel(R) Pentium(R) CPU G3220 @ 3.00GHz
vendor: Intel Corp.
physical id: 1
bus info: cpu@0
size: 3GHz
capacity: 3GHz
width: 64 bits
```

Sección 6.3: Lista de hardware

Ubuntu:

lshw es una pequeña herramienta para extraer información detallada sobre la configuración de hardware de la máquina. Puede informar de la configuración exacta de la memoria, la versión del firmware, la configuración de la placa base, la versión y velocidad de la CPU, la configuración de la caché, la velocidad del bus, etc.

```
$ sudo lshw | less (or more)
$ sudo lshw -html > myhardware.html
$ sudo lshw -xml > myhardware.xml
```

Para mostrar información PCI

```
$ lspci -tv
```

Para ver la información del USB

```
$ lsusb -tv
```

Para ver la información de la BIOS

```
$ dmidecode -q | less
```

Para ver información específica sobre el disco (disco sda en el ejemplo) puede utilizar:

```
$ hdparm -i /dev/sda
```

Algunas utilidades/comandos adicionales ayudarán a recopilar información adicional:

```
$ smartctl -A /dev/sda | grep Power_On_Hours # ¿Cuánto tiempo ha estado encendido este disco (sistema) en total?
$ hdparm -tT /dev/sda # Haz una prueba de velocidad de lectura en el disco sda
$ badblocks -s /dev/sda # Prueba de bloques ilegibles en el disco sda
```

Sección 6.4: Buscar información sobre el modelo y la velocidad de la CPU

Ubuntu:

```
$ cat /proc/cpuinfo
```

Ejemplo de salida:

```
processor : 0
vendor_id : GenuineIntel
cpu family : 6
model : 15
model name : Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz
stepping : 11
cpu MHz : 1596.000
cache size : 4096 KB
physical id : 0
siblings : 4
core id : 0
cpu cores : 4
apicid : 0
initial apicid : 0
fpu : yes
fpu_exception : yes
cpuid level : 10
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc arch_perfmon pebs bts rep_good
pni
dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr pdcm lahf_lm tpr_shadow vnmi flexpriority
bogomips : 4800.18
clflush size : 64
cache_alignment : 64
address sizes : 36 bits physical, 48 bits virtual
power management:
....
..
processor : 3
vendor_id : GenuineIntel
cpu family : 6
model : 15
model name : Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz
stepping : 11
cpu MHz : 1596.000
cache size : 4096 KB
physical id : 0
siblings : 4
core id : 3
cpu cores : 4
apicid : 3
initial apicid : 3
fpu : yes
fpu_exception : yes
cpuid level : 10
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc arch_perfmon pebs bts rep_good
pni
dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr pdcm lahf_lm tpr_shadow vnmi flexpriority
```

```
bogomips : 4800.30
clflush size : 64
cache_alignment : 64
address sizes : 36 bits physical, 48 bits virtual
power management:
```

procesador de recuento (incluidos núcleos):

```
$ grep -c processor /proc/cpuinfo
```

Sección 6.5: Supervisión de procesos y recopilación de información

En general, usted tiene dos maneras de supervisar los procesos en linux host

Monitorización estática

El comando más utilizado es **ps** (es decir, estado del proceso), que se utiliza para proporcionar información sobre los procesos que se están ejecutando actualmente, incluidos sus números de identificación de proceso (PID).

He aquí algunas opciones útiles para recabar información específica.

Listar procesos en una jerarquía

```
$ ps -e -o pid,args -forest
```

Listar procesos ordenados por % de uso de cpu

```
$ ps -e -o pcpu,cpu,nice,state,cputime,args --sort pcpu | sed '/^ 0.0 /d'
```

Lista procesos ordenados por uso de mem (KB).

```
$ ps -e -orss=,args= | sort -b -k1,1n | pr -TW$COLUMNS
```

Listar todos los hilos de un proceso en particular (proceso “firefox-bin” en el ejemplo)

```
$ ps -C firefox-bin -L -o pid,tid,pcpu,state
```

Después de encontrar un proceso específico, puede obtener información relacionada con él usando **lsdf** para listar las rutas que el id del proceso tiene abiertas

```
$ lsdf -p $$
```

O basándose en la ruta encuentre la lista de procesos que tienen abierta la ruta especificada

```
$ lsdf ~
```

Supervisión interactiva

La herramienta más conocida para la supervisión dinámica es:

```
$ top
```

Que en su mayoría por defecto de comandos que tienen gran cantidad de opciones para filtrar y representar la información en tiempo real (en comparación con el comando **ps**).

Sin embargo, hay opciones más avanzadas que pueden considerarse e instalarse como reemplazo de **top**

```
$ htop -d 5
```

o

```
$ atop
```

Que tiene la capacidad de registrar todas las actividades en el archivo de registro (por defecto **atop** registrará toda la actividad en cada 600 segundos) A esta lista hay algunos comandos especializados como **iotop** o **iftop**

```
$ sudo iotop
```

Capítulo 7: comando ls

Sección 7.1: Opciones del comando ls

Lista completa de opciones:

ls -a lista todos los archivos incluyendo los archivos ocultos que empiezan por '.'

ls --color lista coloreada [=always/never/auto]

ls -d lista directorios - con '*'

ls -F añade un carácter de */=>@| a las entradas

ls -i lista el número de índice del inodo del archivo

ls -l lista con formato largo - mostrar permisos

ls -la lista formato largo incluyendo archivos ocultos

ls -lh lista formato largo con tamaño de archivo legible

ls -ls lista con formato largo con tamaño de fichero

ls -r lista en orden inverso

ls -R lista recursivamente el árbol de directorios

ls -s lista el tamaño del archivo

ls -S ordenar por tamaño de archivo

ls -t ordenar por fecha y hora

ls -X ordenar por nombre de extensión

Sección 7.2: Comando ls con las opciones más utilizadas

ls muestra archivos y directorios en el directorio de trabajo actual. (si no se pasan argumentos.) (No muestra los archivos ocultos que empiezan por . por defecto.)

```
user@ubuntu14:/usr$ ls
bin games include lib lib32 local sbin share src
```

Para ver todos los archivos (también los archivos/carpetas ocultas). Utilice **ls -a** O **ls -all**

```
user@ubuntu14:/usr$ ls -a
. .. bin games include lib lib32 local sbin share src
```

Para diferenciar entre archivos y carpetas y enlaces simbólicos y otros, utilice **ls -F** O **ls --classify**

```
user@ubuntu14:~$ ls -F
bash_profile_course chat_apps/ Desktop/ Downloads/ foxitsoftware/
Public/ test/ bin/ ClionProjects/ Documents/ IDE/ Music/
Pictures/ Templates/ Videos/
```

Aquí, los caracteres finales se utilizan para distinguir archivos y carpetas.

[/] sugerir directorio.

[*] sugerir ejecutables.

[@] sugieren enlaces simbólicos.

Para obtener más detalles sobre los archivos y directorios, utilice **ls -l**

```
user@ubuntu14:~/example$ ls -l
total 6464
-rw-r--r-- 1 dave dave 41 Dec 24 12:19 Z.txt
drwxr-xr-x 2 user group 4096 Dec 24 12:00 a_directory
-rw-r--r-- 1 user group 6 Dec 24 12:01 a_file
lrwxrwxrwx 1 user group 6 Dec 24 12:04 a_link -> a_file
-rw-r--r-- 1 user group 6 Dec 24 12:03 a_newer_file
-rw-r----- 1 user group 6586816 Dec 24 12:07 big.zip
```

En este ejemplo, el tamaño total del contenido es de 6460KB.

A continuación, hay una entrada para cada archivo/directorio en orden alfabético, con las mayúsculas antes que las minúsculas.

El primer carácter es el tipo (por ejemplo, **d** - directorio, **l** - enlace).

Los siguientes 9 caracteres muestran los permisos para el usuario, grupo y otros.

A continuación, se indica el número de enlaces duros y, por último, el nombre y el grupo del propietario.

El siguiente campo es el tamaño en bytes. Esto se puede mostrar de forma amigable añadiendo la opción **-h**, por ejemplo, 6586816 se muestra como 6.3M

A continuación, aparece una marca de tiempo (normalmente la hora de modificación).

El último campo es el nombre. Nota: los enlaces también muestran el destino del enlace.

Capítulo 8: Compresión de archivos con el comando 'tar'

Opciones comunes

<code>-c --create</code>	- Crea un nuevo archivo.
<code>-x --extract</code>	Extraer ficheros de un archivo.
<code>-t --list</code>	Listar el contenido de un archivo.
<code>-f --file=ARCHIVE</code>	Utilizar archivo o dir ARCHIVE.
<code>-v --verbose</code>	Lista detallada de los archivos procesados.

Opciones de compresión

<code>-a --auto-compress</code>	- Utilice el sufijo de archivo para determinar el programa de compresión.
<code>-j --bzip2</code>	Filtra el archivo a través de bzip2.
<code>-J --xz --lzma</code>	Filtra el archivo a través de xz.
<code>-z --gzip</code>	Filtra el archivo mediante gzip.

Sección 8.1: Comprimir una carpeta

Esto crea un archivo simple de una carpeta:

```
tar -cf ./my-archive.tar ./my-folder/
```

La salida detallada muestra qué ficheros y directorios se añaden al archivo; utilice la opción `-v`:

```
tar -cvf ./my-archive.tar ./my-folder/
```

Para archivar una carpeta comprimida con 'gzip', hay que utilizar la opción `-z`:

```
tar -czf ./my-archive.tar.gz ./my-folder/
```

En su lugar, puede comprimir el archivo con 'bzip2', utilizando la opción `-j`:

```
tar -cjf ./my-archive.tar.bz2 ./my-folder/
```

O comprimir con 'xz', utilizando la opción `-J`:

```
tar -cJf ./my-archive.tar.xz ./my-folder/
```

Sección 8.2: Extraer una carpeta de un archivo

Hay un ejemplo para extraer una carpeta de un archivo en la ubicación actual:

```
tar -xf archive-name.tar
```

Si desea extraer una carpeta de un archivo a un destino específico:

```
tar -xf archive-name.tar -C ./directory/destination
```

Sección 8.3: Listar el contenido de un archivo

Listar el contenido de un archivo sin extraerlo:

```
tar -tf archive.tar.gz
Folder-In-Archive/
Folder-In-Archive/file1
Folder-In-Archive/Another-Folder/
Folder-In-Archive/Another-Folder/file2
```


Sección 8.4: Listar el contenido del archivo

Hay un ejemplo de listado de contenidos:

```
tar -tvf archive.tar
```

La opción `-t` se utiliza para el listado. Para listar el contenido de un archivo tar.gz, hay que seguir utilizando la opción `-z`:

```
tar -tzvf archive.tar.gz
```

Sección 8.5: Comprimir y excluir una o varias carpetas

Si desea extraer una carpeta, pero quiere excluir una o varias carpetas durante la extracción, puede utilizar la opción `--exclude`.

```
tar -cf archive.tar ./my-folder/ --exclude="my-folder/sub1" --exclude="my-folder/sub3"
```

Con este árbol de carpetas:

```
my-folder/  
  sub1/  
  sub2/  
  sub3/
```

El resultado será:

```
./archive.tar  
  my-folder/  
    sub2/
```

Sección 8.6: Componentes principales de la banda

Para eliminar cualquier número de componentes principales, utilice la opción `--strip-components`:

```
--strip-components=NUMBER  
strip NUMBER leading components from file names on extraction
```

Por ejemplo, para eliminar la carpeta inicial, utilice:

```
tar -xf --strip-components=1 archive-name.tar
```

Capítulo 9: Servicios

Sección 9.1: Lista de servicios en ejecución en Ubuntu

Para obtener una lista del servicio en su sistema, puede ejecutar:

```
service --status-all
```

La salida de `service --status-all` lista el estado de los servicios controlados por System V.

El `+` indica que el servicio se está ejecutando, `-` indica un servicio detenido. Puede comprobarlo ejecutando `service SERVICENAME status` para un servicio `+` y `-`.

Algunos servicios son gestionados por **Upstart**. Puede comprobar el estado de todos los servicios Upstart con `sudo initctl list`. Cualquier servicio gestionado por Upstart también se mostrará en la lista proporcionada por `service --status-all` pero se marcará con una `?`.

ref: <https://askubuntu.com/questions/407075/how-to-read-service-status-all-results>

Sección 9.2: Gestión de servicios Systemd

Listado de servicios

- `systemctl` Para listar los servicios en ejecución.
- `systemctl --failed` Para listar los servicios fallidos.

Gestión de objetivos (similar a los niveles de ejecución en SysV)

- `systemctl get-default` Para encontrar el objetivo por defecto de su sistema.
- `systemctl set-default <target-name>` Para establecer el destino predeterminado de su sistema.

Gestión de servicios en tiempo de ejecución

- `systemctl start [service-name]` Para iniciar un servicio.
- `systemctl stop [service-name]` Para detener un servicio.
- `systemctl restart [service-name]` Para reiniciar un servicio.
- `systemctl reload [service-name]` Para solicitar al servicio que recargue su configuración.
- `systemctl status [service-name]` Para mostrar el estado actual de un servicio.

Gestión del inicio automático de servicios

- `systemctl is-enabled [service-name]` Para mostrar si un servicio está habilitado en el arranque del sistema.
- `systemctl is-active [service-name]` Para mostrar si un servicio está actualmente activo (en ejecución).
- `systemctl enable [service-name]` Para activar un servicio en el arranque del sistema.
- `systemctl disable [service-name]` Para desactivar un servicio en el arranque del sistema.

Servicios de enmascaramiento

- `systemctl mask [service-name]` Para enmascarar un servicio (dificulta el inicio de un servicio por error).
- `systemctl unmask [service-name]` Para desenmascarar un servicio.

Reiniciar systemd

```
systemctl daemon-reload
```

Capítulo 10: Gestión de servicios

Sección 10.1: Diagnosticar un problema con un servicio

En sistemas que utilizan `systemd`, como Fedora => 15, Ubuntu (Server y Desktop) >= 15.04, y RHEL/CentOS >= 7:

```
systemctl status [servicename]
```

...donde `[servicename]` es el servicio en cuestión; por ejemplo, `systemctl status sshd`.

Esto mostrará la información básica de estado y cualquier error reciente registrado.

Puedes ver más errores con `journalctl`. Por ejemplo, `journalctl -xe` cargará los últimos 1000 registrados en un paginador (como `less`), saltando hasta el final. También puede utilizar `journalctl -f`, que seguirá los mensajes de registro a medida que entran.

Para ver los registros de un servicio concreto, utilice el indicador `-t`, como se indica a continuación:

```
journalctl -f -t sshd
```

Otras opciones útiles incluyen `-p` para prioridad (`-p warnings` para ver sólo advertencias y superiores), `-b` para "desde el último arranque", y `-S` para "desde" - juntando todo esto, podríamos hacer.

```
journalctl -p err -S yesterday
```

para ver todos los elementos registrados como errores desde ayer.

Si `journalctl` no está disponible, o si está siguiendo registros de errores de aplicaciones que no utilizan el diario del sistema, el comando `tail` puede utilizarse para mostrar las últimas líneas de un archivo. Una bandera útil para `tail` es `-f` (para "seguir"), que hace que `tail` continúe mostrando los datos a medida que se añaden al archivo. Para ver los mensajes de la mayoría de los servicios del sistema:

```
tail -f /var/log/messages
```

O, si el servicio es privilegiado y puede registrar datos sensibles:

```
tail -f /var/log/secure
```

Algunos servicios tienen sus propios archivos de registro, un buen ejemplo es `auditd`, el demonio de auditoría de Linux, que tiene sus registros almacenados en `/var/log/audit/`. Si no ve la salida de su servicio en `/var/log/messages`, intente buscar registros específicos del servicio en `/var/log/`.

Sección 10.2: Iniciar y detener servicios

En sistemas que utilizan scripts de inicio del tipo System-V, como RHEL/CentOS 6:

```
service <service> start
```

```
service <service> stop
```

En sistemas que utilizan `systemd`, como Ubuntu (Server y Desktop) >= 15.04, y RHEL/CentOS >= 7:

```
systemctl <service> dnsmasq
```

```
systemctl <service> dnsmasq
```

Sección 10.3: Obtener el estado de un servicio

En sistemas que utilizan scripts de inicio del tipo System-V, como RHEL/CentOS 6:

```
service <service> status
```

En sistemas que utilizan `systemd`, como Ubuntu (Server y Desktop) ≥ 15.04 , y RHEL/CentOS ≥ 7.0 :

```
systemctl status <service>
```

Capítulo 11: Modificación de usuarios

Parámetro	Detalles
username	El nombre del usuario. No use mayúsculas, no use puntos, no lo termine en guión, no debe incluir dos puntos, ni caracteres especiales. No puede empezar por un número.

Sección 11.1: Establecer su propia contraseña

passwd

Sección 11.2: Establecer la contraseña de otro usuario

Ejecute lo siguiente como root:

passwd username

Sección 11.3: Añadir un usuario

Ejecute lo siguiente como root:

useradd username

Sección 11.4: Eliminar un usuario

Ejecute lo siguiente como root:

userdel username

Sección 11.5: Eliminar un usuario y su carpeta de inicio

Ejecute lo siguiente como root:

userdel -r username

Sección 11.6: Listado de grupos a los que pertenece el usuario actual

groups

Puede encontrar información más detallada sobre los ID numéricos de usuario y grupo con el comando **id**.

Sección 11.7: Listado de grupos a los que pertenece un usuario

groups username

Puede encontrar información más detallada sobre los identificadores numéricos de usuario y grupo con **id username**.

Capítulo 12: LAMP Stack

LAMP (**L**inux **A**pache **M**ySQL **P**HP) se compone del sistema operativo Linux como entorno de desarrollo, el servidor HTTP Apache como servidor web, el sistema de gestión de bases de datos relacionales MySQL (RDBMS) como sistema de base de datos y el lenguaje de programación PHP como lenguaje de programación del lado del servidor (back-end).

LAMP se utiliza como una solución de pila de tecnologías de código abierto para el área de desarrollo web. La versión para Windows de esta pila se llama WAMP (**W**indows **A**pache **M**ySQL **P**HP).

Sección 12.1: Instalación de LAMP en Arch Linux

Con esta línea instalaremos todos los paquetes necesarios en un solo paso, y la última actualización:

```
pacman -Syu apache php php-apache mariadb
```

HTTP

Editar

```
/etc/httpd/conf/httpd.conf
```

Cambie `ServerAdmin` `you@example.com` según sus necesidades.

La carpeta de las Páginas WEB por defecto es `ServerRoot "/etc/httpd"`. El directorio debe establecerse en la misma carpeta, así que cambie la línea.

```
<Directory "/etc/httpd">
```

Esta carpeta debe tener acceso de lectura y ejecución, por lo que

```
chmod o+x /etc/httpd
```

Cambie `AllowOverride` from `none` (default) to `All` para que el `.htaccess` funcione.

Ahora necesitas la carpeta `~/public_html` para cada usuario. (para obtener la página raíz de cada usuario como `http://localhost/~yourusername/`. Desmarque esta línea:

```
Include conf/extra/httpd-userdir.conf
```

Ahora como root necesitas crear el `~/public_html` para cada usuario y cambiar el acceso a (755) de cada uno.

```
chmod 755 /home
chmod 755 /home/username
chmod 755 /home/username/public_html
```

Puede comentar esta línea si desea utilizar SSL:

```
LoadModule ssl_module modules/mod_ssl.so
```

Si necesita utilizar dominios virtuales, descomente la línea:

```
Include conf/extra/httpd-vhosts.conf
```

y en `/etc/httpd/conf/extra/httpd-vhosts.conf` debes añadir todos los dominios virtuales. (además en `/etc/hosts` si quieres probar esos dominios virtuales)

Edita `/etc/httpd/conf/extra/httpd-default.conf` y cambie `ServerSignature` a `Off` y `ServerToken` a `Prod` para ocultar datos críticos.

PHP

Edita: `/etc/httpd/conf/httpd.conf`

Comentar: `LoadModule mpm_event_module modules/mod_mpm_event.so`

Descomentar: `LoadModule mpm_prefork_module modules/mod_mpm_prefork.so`

Como último elemento de la lista `LoadModule`, añade `LoadModule php7_module modules/libphp7.so`

Como último elemento de la lista `Include`, añade `Include conf/extra/php7_module.conf`

Editar `/etc/php/php.ini`

Descomentar `extension=mysqli.so` and `extension=pdo_mysql.so`

Cambia la zona horaria según necesites, por ejemplo:

```
date.timezone = America/Argentina/Buenos_Aires, date.default_latitude = 0.0,  
date.default_longitude = 0.0
```

MySQL

Ejecutar como root:

```
mysql_install_db --user=mysql --basedir=/usr --datadir=/var/lib/mysql
```

Ahora tienes la raíz del servidor MySQL.

Iniciar el demonio MySQL:

```
systemctl enable mysqld  
systemctl start mysqld
```

Por fin, corre:

```
sh /usr/bin/mysql_secure_installation
```

Todo ello para conseguir un servidor web listo para ser personalizado como necesites.

Sección 12.2: Instalación de LAMP en Ubuntu

Instalar apache:

```
sudo apt-get install apache2
```

Instalar MySQL:

```
sudo apt-get install mysql-server
```

Instalar PHP:

```
sudo apt-get install php libapache2-mod-php
```

Reinicia apache2:

```
sudo systemctl restart apache2
```

Compruebe la instalación de PHP:

```
php -r 'echo "\n\nYour PHP installation is working fine.\n\n\n";'
```

Sección 12.3: Instalación de LAMP en CentOS

Instalar el servidor web Apache

El primer paso es instalar el servidor web Apache.

```
sudo yum -y install httpd
```

Una vez instalado, active (para que se ejecute al inicio) e inicie el servicio de servidor web Apache.

```
sudo systemctl enable --now httpd
```

Dirija su navegador a

<http://localhost>

Verá la página por defecto del servidor web Apache.

Instalar el servidor MariaDB

El segundo paso es instalar MariaDB:

```
sudo yum -y install mariadb-server
```

A continuación, inicie y habilite (al iniciar) el servidor MariaDB:

```
sudo systemctl enable --now mariadb
```

Si es necesario, utilice **mysql_secure_installation** para asegurar su base de datos.

Este script le permitirá hacer lo siguiente:

- Cambiar la contraseña del usuario root
- Eliminar bases de datos de prueba
- Desactivar el acceso remoto

Instalar PHP

```
sudo yum -y install php php-common
```

A continuación, reinicie el servicio httpd de Apache.

```
sudo systemctl restart httpd
```

Para probar PHP, cree un archivo llamado **index.php** en **/var/www/html**.

A continuación, añada la siguiente línea al archivo:

A continuación, dirija su navegador a:

<http://localhost/index.php>

Debería ver información relacionada con su servidor. Si no es así, asegúrese de que php está instalado correctamente ejecutando el siguiente comando:

```
php --version
```

Si recibes algo como:

```
PHP 5.4.16 (cli) (built: Nov 6 2016 00:29:02) Copyright (c) 1997-2013 The PHP Group
```

Entonces PHP está instalado correctamente. Si este es el caso, por favor asegúrese de que ha reiniciado su servidor web.

Capítulo 13: comando tee

Opciones

-a, --append
-i, --ignore-interrupts
--help
--version

Descripción

Añadir a los ARCHIVOS dados. No sobrescribir.
Ignora las señales de interrupción.
Mostrar un mensaje de ayuda y salir.
Mostrar información sobre la versión y salir.

tee - leer de la entrada estándar y escribir en la salida estándar y archivos.

El comando **tee** debe su nombre al separador en T de la fontanería, que divide el agua en dos direcciones y tiene forma de T mayúscula.

tee copia datos de la entrada estándar a cada ARCHIVO, y también a la salida estándar. En efecto, **tee** duplica su entrada, dirigiéndola a múltiples salidas a la vez.

Sección 13.1: Escribir la salida a stdout, y también a un archivo

El siguiente comando muestra la salida sólo en la pantalla (stdout).

```
$ ls
```

El siguiente comando escribe la salida sólo en el archivo y no en la pantalla.

```
$ ls > file
```

El siguiente comando (con la ayuda del comando **tee**) escribe la salida tanto en la pantalla (stdout) como en el archivo.

```
$ ls | tee file
```

Sección 13.2: Escribir la salida desde el medio de una cadena de tuberías a un archivo y pasarlo de nuevo a la tubería

También puede utilizar el comando **tee** para almacenar la salida de un comando en un archivo y redirigir la misma salida a otro comando.

El siguiente comando escribirá las entradas crontab actuales en un archivo `crontab-backup.txt` y pasará las entradas crontab al comando **sed**, que realizará la sustitución. Después de la sustitución, se añadirá como una nueva tarea cron.

```
$ crontab -l | tee crontab-backup.txt | sed 's/old/new/' | crontab -
```

Sección 13.3: Escribir la salida en varios archivos

Puede enviar la salida a varios archivos (incluido el terminal) utilizando **tee** de la siguiente forma:

```
$ ls | tee file1 file2 file3
```

Sección 13.4: Indica al comando tee que añada al archivo

Por defecto, el comando **tee** sobrescribe el archivo. Puede ordenar a **tee** que añada datos al archivo mediante la opción **-a**, como se muestra a continuación.

```
$ ls | tee -a file
```

Capítulo 14: Secure Shell (SSH)

Un shell seguro se utiliza para acceder remotamente a un servidor desde un cliente a través de una conexión cifrada. OpenSSH se utiliza como alternativa a las conexiones Telnet que consiguen un acceso shell remoto pero sin cifrar. El cliente OpenSSH está instalado por defecto en la mayoría de las distribuciones GNU/Linux y se utiliza para conectarse a un servidor. Estos ejemplos muestran cómo utilizar la suite SSH para aceptar conexiones SSH y conectarse a otro host.

Sección 14.1: Conectarse a un servidor remoto

Para conectarnos a un servidor debemos utilizar `ssh` en el cliente de la siguiente manera,

```
# ssh -p puerto usuario@direccion-servidor
```

- **puerto** - El puerto ssh de escucha del servidor (puerto 22 por defecto).
- **usuario** - Debe ser un usuario existente en el servidor con privilegios SSH.
- **direccion-servidor** - La IP/Dominio del servidor.

Para poner un ejemplo real, imaginemos que estás creando una página web. La compañía que elegiste para alojar tu sitio te dice que el servidor está localizado en **web-servers.com** en un puerto personalizado de **2020** y tu nombre de cuenta **usr1** ha sido elegido para crear un usuario en el servidor con privilegios SSH. En este caso el comando `ssh` utilizado sería el siguiente

```
# ssh -p 2020 usr1@web-servers.com
```

Si el nombre de cuenta en el sistema remoto es el mismo que el del cliente local, puede omitir el nombre de usuario. Así que, si usted es `usr1` en ambos sistemas, entonces usted puede simplemente utilizar `web-servers.com` en lugar de `usr1@web-servers.com`.

Cuando no pueda acceder directamente a un servidor al que desea conectarse, puede intentar utilizar el conmutador ProxyJump para conectarse a él a través de otro servidor al que pueda acceder y que pueda conectarse al servidor deseado.

```
# ssh -J usr1@10.0.0.1:2020 usr2@10.0.0.2 -p 2222
```

Esto le permitirá conectarse al servidor `10.0.0.2` (ejecutando `ssh` en el puerto `2222`) a través del servidor en `10.0.0.1` (ejecutando `ssh` en el puerto `2020`). Por supuesto, necesitarás tener cuentas en ambos servidores. Ten en cuenta también que el parámetro `-J` se introdujo en la versión 7.3 de OpenSSH.

Sección 14.2: Instalación del paquete OpenSSH

Tanto la conexión a un servidor SSH eliminado como la aceptación de conexiones SSH requieren la instalación de openssh

Debian:

```
# apt-get install openssh
```

Arch Linux:

```
# pacman -S openssh
```

Yum:

```
# yum install openssh
```

Sección 14.3: Configurar un servidor SSH para que acepte conexiones

Primero debemos editar el archivo de configuración del demonio SSH. Aunque en diferentes distribuciones de Linux puede estar localizado en diferentes directorios, normalmente se almacena en `/etc/ssh/sshd_config`.

Utilice su editor de texto para cambiar los valores establecidos en este archivo, todas las líneas que comienzan con # se comentan y deben tener este carácter eliminado para tener algún efecto. Una lista de recomendaciones sigue como tal.

```
Port (chose a number between 0 - 65535, normally greater than four digits)
PasswordAuthentication yes
AllowUsers      user1 user2 ...etc
```

Tenga en cuenta que es preferible desactivar los inicios de sesión con contraseña y utilizar claves SSH para mejorar la seguridad, como se explica en este documento.

Sección 14.4: Conexión sin contraseña (mediante un par de claves)

En primer lugar, necesitarás un par de claves. Si aún no tienes uno, echa un vistazo al tema 'Generar clave pública y privada'.

Tu par de claves está compuesto por una clave privada (id_rsa) y una clave pública (id_rsa.pub). Todo lo que tienes que hacer es copiar la clave pública en el host remoto y añadir su contenido al archivo `~/.ssh/authorized_keys`.

Una forma sencilla de hacerlo es:

```
ssh <user>@<ssh-server> 'cat >> ~/.ssh/authorized_keys' < id_rsa.pub
```

Una vez que la clave pública esté colocada correctamente en el directorio principal de su usuario, sólo tiene que iniciar sesión utilizando la clave privada correspondiente:

```
ssh <user>@<ssh-server> -i id_rsa
```

Sección 14.5: Generar clave pública y privada

Para generar claves para el cliente SSH:

```
ssh-keygen [-t rsa | rsa1 | dsa ] [-C <comment>] [-b bits]
```

Por ejemplo:

```
ssh-keygen -t rsa -b 4096 -C myemail@email.com
```

La ubicación por defecto es `~/.ssh/id_rsa` para la clave privada y `~/.ssh/id_rsa.pub` para la pública.

Para más información, visite man.openbsd.org

Sección 14.6: Desactivar el servicio ssh

Esto desactivará el servicio del lado del servidor SSH, ya que si es necesario esto asegurará que los clientes no puedan conectarse a través de **ssh**.

Ubuntu:

```
sudo service ssh stop
sudo systemctl disable sshd.service
```

Debian:

```
sudo /etc/init.d/ssh stop
sudo systemctl disable sshd.service
```

Arch Linux:

```
sudo killall sshd
sudo systemctl disable sshd.service
```

Capítulo 15: SCP

Sección 15.1: Copia segura

El comando `scp` se utiliza para copiar de forma segura un archivo hacia o desde un destino remoto. Si el archivo se encuentra directamente en el sistema de trabajo actual, basta con el nombre del archivo; de lo contrario, se requiere una ruta completa que incluya el nombre del host remoto, por ejemplo, `remote_user@some_server.org:/path/to/file`.

Copie el archivo local en su CWD al nuevo directorio

```
scp localfile.txt /home/friend/share/
```

Copiar el archivo remoto al directorio de trabajo actual

```
scp rocky@arena51.net:/home/rocky/game/data.txt ./
```

Copiar archivo de una ubicación remota a otra ubicación remota

```
scp mars@universe.org:/beacon/light/bitmap.conf jupiter@universe.org:/beacon/night/
```

Para copiar directorios y subdirectorios utilice la opción recursiva `-r` de scp

```
scp -r user@192.168.0.4:~/project/* ./workspace/
```

Sección 15.2: Uso básico

Copiar archivo remoto en directorio local

```
scp user@remotehost.com:/remote/path/to/foobar.md /local/dest
```

Copiar archivo local a directorio remoto

```
scp foobar.md user@remotehost.com:/remote/dest
```

Se pueden utilizar archivos de claves (igual que ssh)

```
scp -i my_key.pem foobar.md user@remotehost.com:/remote/dest
```

Capítulo 16: GnuPG (GPG)

GnuPG es un sofisticado sistema de gestión de claves que permite firmar o cifrar datos de forma segura. GPG es una herramienta de línea de comandos utilizada para crear y manipular claves GnuPG.

GnuPG es el más utilizado para tener conexiones SSH (Secure Shell) sin contraseña o cualquier medio de autenticación interactiva, lo que mejora el nivel de seguridad significativamente.

Las siguientes secciones describen las formas de crear, utilizar y mantener la seguridad de las claves GnuPG.

Sección 16.1: Exportar la clave pública

Para que tu par de claves pública-privada sea útil, debes poner tu clave pública a libre disposición de los demás. Asegúrate de que estás trabajando con tu clave pública, ya que nunca debes compartir tu clave privada. Puedes exportar tu clave pública con el siguiente comando:

```
gpg --armor --export EMAIL_ADDRESS > public_key.asc
```

donde `EMAIL_ADDRESS` es la dirección de correo electrónico asociada a la clave

Alternativamente, puedes subir tu clave pública a un servidor de claves públicas como `keys.gnupg.net` para que otros puedan utilizarla. Para ello, introduce lo siguiente en un terminal:

```
gpg --list-keys
```

A continuación, busque la cadena de 8 dígitos (el ID primario) asociada a la clave que desea exportar. A continuación, emita el comando:

```
gpg --send-keys PRIMARY_ID
```

donde `PRIMARY_ID` es el ID real de esa clave.

Ahora, la clave pública se ha subido al servidor de claves y está a disposición del público.

Sección 16.2: Crear y utilizar rápidamente una clave GnuPG

Instala `haveged` (ejemplo `sudo apt-get install haveged`) para acelerar el proceso de bytes aleatorios. Entonces:

```
gpg --gen-key
gpg --list-keys
```

salidas:

```
pub 2048R/NNNNNNNN 2016-01-01
uid Name <name@example.com>
sub 2048R/xxxxxxxx 2016-01-01
```

Pues publícalo:

```
gpg --keyserver pgp.mit.edu --send-keys NNNNNNNN
```

A continuación, plan para revocar: <https://www.hackdiary.com/2004/01/18/revoking-a-gpg-key/>

Capítulo 17: Configuración de red

Este documento cubre los aspectos básicos de las redes TCP/IP, la administración de redes y la configuración del sistema. Linux puede soportar múltiples dispositivos de red. Los nombres de los dispositivos están numerados y comienzan en cero y cuentan hacia arriba. Por ejemplo, un ordenador con dos NICs tendrá dos dispositivos etiquetados como `eth0` y `eth1`.

Sección 17.1: Resolución DNS local

Archivo: `/etc/hosts` contiene una lista de hosts que deben ser resueltos localmente (no por DNS)

Ejemplo del contenido del archivo:

```
127.0.0.1          your-node-name.your-domain.com localhost.localdomain    localhost
XXX.XXX.XXX.XXX   node-name
```

El formato del fichero hosts está especificado en el [RFC 952](#).

Sección 17.2: Configurar servidores DNS para la resolución de nombres de dominio

Archivo: `/etc/resolv.conf` contiene una lista de servidores DNS para la resolución de nombres de dominio

Ejemplo del contenido del archivo:

```
nameserver 8.8.8.8    # Dirección IP del servidor de nombres primario
nameserver 8.8.4.4    # Dirección IP del servidor de nombres secundario
```

En el caso de un servidor DNS interno, puede validar si este servidor resuelve los nombres DNS correctamente utilizando el comando `dig`:

```
$ dig google.com @your.dns.server.com +short
```

Sección 17.3: Ver y manipular rutas

Manipular la tabla de enrutamiento IP utilizando `route`

Mostrar tabla de enrutamiento

```
$ route           # Muestra la lista o las rutas y también resuelve los nombres de host
$ route -n        # Muestra la lista de rutas sin resolver los nombres de host para obtener
                  resultados más rápidos.
```

Añadir/Borrar ruta

Opción	Descripción
<code>add</code> o <code>del</code>	Añadir o eliminar una ruta
<code>-host x.x.x.x</code>	Añadir ruta a un único host identificado por la dirección IP
<code>-net x.x.x.x</code>	Añadir ruta a una red identificada por la dirección de red
<code>gw x.x.x.x</code>	Especifique la puerta de enlace de la red
<code>netmask x.x.x.x</code>	Especifique la máscara de red
<code>default</code>	Añadir una ruta por defecto

Ejemplos:

- añadir ruta a un host `$ route add -host x.x.x.x eth1`
- añadir ruta a una red `$ route add -net 2.2.2.0 netmask 255.255.255.0 eth0`
- Alternativamente, también puede utilizar el formato cidr para añadir una ruta a la red `route add -net 2.2.2.0/24 eth0`

- añadir puerta de enlace por defecto `$ route add default gw 2.2.2.1 eth0`
- eliminar una ruta `$ route del -net 2.2.2.0/24`

Manipular la tabla de enrutamiento IP utilizando ip

Mostrar tabla de enrutamiento

`$ ip route show # List routing table`

Añadir/Borrar ruta

Opción	Descripción
add o del o change o append o replace	Cambiar una ruta
show o flush	el comando muestra el contenido de las tablas de enrutamiento o lo elimina
restore	restaurar la información de la tabla de enrutamiento desde stdin
get	este comando obtiene una única ruta a un destino e imprime su contenido exactamente como lo ve el kernel

Ejemplos:

- Establecer la puerta de enlace predeterminada en 1.2.3.254 `$ ip route add default via 1.2.3.254`
- Añade una ruta por defecto (para todas las direcciones) a través de la puerta de enlace local 192.168.1.1 que se puede alcanzar en el dispositivo eth0 `$ ip route add default via 192.168.1.1 dev eth0`

Sección 17.4: Configurar un nombre de host para otro sistema de la red

Puede configurar su sistema Linux (o macOS) para vincular un identificador <hostname> a la dirección IP de algún otro sistema de su red. Puede configurarlo:

En todo el sistema. Debe modificar el archivo /etc/hosts. Sólo tienes que añadir a ese archivo una nueva línea que contenga:

1. la dirección IP del sistema remoto <ip_rem>,
2. uno o más espacios en blanco, y
3. el identificador <hostname>.

Para un solo usuario. Deberá modificar el archivo ~/.hosts --- tendrá que crearlo. No es tan sencillo como para todo el sistema. Aquí puedes ver una explicación.

Por ejemplo, puedes añadir esta línea utilizando la herramienta cat de Unix. Supongamos que quieres hacer un ping a un PC en tu red local cuya dirección IP es 192.168.1.44 y quieres referirte a esa dirección IP sólo por remote_pc. Entonces debes escribir en tu shell:

```
$ sudo cat 192.168.1.44 remote_pc
```

Entonces usted puede hacer que el ping sólo por:

```
$ ping remote_pc
```

Sección 17.5: Detalles de la interfaz

ifconfig

Enumerar todas las interfaces disponibles en la máquina

```
$ ifconfig -a
```

Enumerar los detalles de una interfaz específica

Sintaxis: `$ ifconfig <interface>`

Ejemplo:

```
$ ifconfig eth0
eth0      Link encap:Ethernet HWaddr xx:xx:xx:xx:xx:xx
          inet addr:x.x.x.x Bcast:x.x.x.x Mask:x.x.x.x
          inet6 addr: xxxx:xxx:xxx:xxx:xxx/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:4426618 errors:0 dropped:1124 overruns:0 frame:0
          TX packets:189171 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:382611580 (382.6 MB) TX bytes:36923665 (36.9 MB)
          Interrupt:16 Memory:fb5e0000-fb600000
```

Ethtool - consulta el controlador de red y la configuración del hardware

Sintaxis `$ ethtool <interface>`

Ejemplo:

```
$ ethtool eth0
Settings for eth0:
Supported ports: [ TP ]
Supported link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
1000baseT/Full
Supported pause frame use: No
Supports auto-negotiation: Yes
Advertised link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
1000baseT/Full
Advertised pause frame use: No
Advertised auto-negotiation: Yes
Speed: 1000Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
MDI-X: on (auto)
Supports Wake-on: pumbg
Wake-on: g
Current message level: 0x00000007 (7)
drv probe link
Link detected: yes
```

ip - mostrar / manipular enrutamiento, dispositivos, políticas de enrutamiento y túneles

Sintaxis: `$ ip { link | ... | route | macsec }` (consulte `man ip` para ver la lista completa de objetos)

Ejemplos

Lista de interfaces de red

```
$ ip link show
```

Cambiar el nombre de la interfaz `eth0` a `wan`

```
$ ip link set dev eth0 name wan
```

Subir (o bajar) la interfaz `eth0`

```
$ ip link set dev eth0 up
```


Lista de direcciones para interfaces

```
$ ip addr show
```

Añadir (o eliminar) ip y máscara (255.255.255.0)

```
$ ip addr add 1.2.3.4/24 brd + dev eth0
```

Sección 17.6: Añadir IP a una interfaz

La dirección IP de una interfaz puede obtenerse mediante DHCP o asignación estática.

DHCP Si está conectado a una red con un servidor DHCP en ejecución, el comando `dhclient` puede obtener una dirección IP para su interfaz.

```
$ dhclient <interface>
```

o alternatively, puedes hacer un cambio en el archivo `/etc/network/interfaces` para que la interfaz se active al arrancar y obtenga la IP DHCP.

```
auto eth0
iface eth0 inet dhcp
```

Configuración estática (cambio permanente) mediante el archivo `/etc/network/interfaces`

Si desea configurar estáticamente los ajustes de la interfaz (cambio permanente), puede hacerlo en el archivo `/etc/network/interfaces`.

Ejemplo:

```
auto eth0 # Activar la interfaz en el arranque
iface eth0 inet static
    address 10.10.70.10
    netmask 255.255.0.0
    gateway 10.10.1.1
    dns-nameservers 10.10.1.20
    dns-nameservers 10.10.1.30
```

Estos cambios persisten incluso después de reiniciar el sistema.

Configuración estática (cambio temporal) mediante la utilidad `ifconfig`

Se puede añadir una dirección IP estática a una interfaz utilizando la utilidad `ifconfig` de la siguiente manera

```
$ ifconfig <interface> <ip-address>/<mask> up
```

Ejemplo:

```
$ ifconfig eth0 10.10.50.100/16 up
```

Capítulo 18: Midnight Commander

Midnight Commander o mc es un gestor de archivos de consola. Este tema incluye la descripción de sus funcionalidades y ejemplos y consejos de cómo usarlo en todo su potencial.

Sección 18.1: Teclas de función del Midnight Commander en modo navegación

Aquí hay una lista de acciones que pueden ser activadas en el modo de exploración del sistema de archivos de Midnight Commander usando las teclas de función de tu teclado.

- F1** Muestra la ayuda
- F2** Abre el menú de usuario
- F3** Muestra el contenido del archivo seleccionado
- F4** Abre el archivo seleccionado en el editor de archivos interno
- F5** Copia el archivo seleccionado en el directorio abierto en el segundo panel
- F6** Mueve el archivo seleccionado al directorio abierto en el segundo panel
- F7** Crea un nuevo directorio en el directorio abierto en el panel actual
- F8** Elimina el archivo o directorio seleccionado
- F9** Se centra en el menú principal de la parte superior de la pantalla.
- F10** Sal de mc

Sección 18.2: Teclas de función de Midnight Commander en modo de edición de archivos

Midnight Commander tiene un editor incorporado que se inicia con la tecla de función **F4** cuando se encuentra sobre el archivo deseado en el modo browse. También puede invocarse en modo autónomo ejecutando

mcedit <filename>

Esta es una lista de acciones que pueden activarse en el modo de edición.

- F1** Muestra la ayuda
- F2** Guarda el archivo actual
- F3** Marca el inicio de la selección de texto. Mueva el cursor en cualquier dirección para seleccionar. La segunda pulsación marca el final de la selección.
- F4** Abre el cuadro de diálogo de búsqueda/sustitución de texto
- F5** Copia el texto seleccionado en la posición del cursor (copiar/pegar)
- F6** Desplaza el texto seleccionado a la posición del cursor (cortar/pegar)
- F7** Abre el cuadro de diálogo de búsqueda de texto
- F8** Borra el texto seleccionado
- F9** Se centra en el menú principal de la parte superior de la pantalla.
- F10** Sal del editor

Capítulo 19: Cambiar de root (chroot)

Cambiar raíz (**chroot**) es una operación que cambia el directorio raíz aparente para el proceso en ejecución actual y sus hijos. Un programa que se ejecuta en un entorno modificado de este tipo no puede acceder a archivos y comandos fuera de ese árbol de directorios del entorno.

Sección 19.1: Requisitos

- privilegios de root.
- otro entorno Linux operativo, como el arranque desde Live CD o una distribución existente.
- coincidencia de las arquitecturas de entorno de origen y destino de **chroot** (compruebe la arquitectura de entorno actual con **uname -m**)
- los módulos del kernel que pueda necesitar en un entorno **chroot** deben cargarse (por ejemplo, con **modprobe**)

Sección 19.2: Cambiar manualmente root de un directorio

1. Asegúrese de que cumple todos los requisitos, según Requisitos
2. Montar los sistemas de archivos temporales de la API:

```
cd /location/of/new/root
mount -t proc proc proc/
mount --rbind /sys sys/
mount --rbind /dev dev/
mount --rbind /run run/ (optionally)
```

3. Si necesita utilizar una conexión a Internet en el entorno **chroot**, copie los detalles DNS:

```
cp /etc/resolv.conf etc/resolv.conf
```

4. Cambia root a /ubicación/de/nuevo/root, especificando el shell (**/bin/bash** en este ejemplo):

```
chroot /location/of/new/root /bin/bash
```

5. Después del chrooting puede ser necesario cargar la configuración local de bash:

```
source /etc/profile
source ~/.bashrc
```

6. Opcionalmente, crea un prompt único para poder diferenciar tu entorno **chroot**:

```
export PS1="(chroot) $PS1"
```

7. Cuando termines con el **chroot**, puedes salir de él mediante:

```
exit
```

8. Desmonta los sistemas de archivos temporales:

```
cd /
umount --recursive /location/of/new/root
```

Sección 19.3: Razones para utilizar chroot

El cambio de root se realiza habitualmente para llevar a cabo el mantenimiento del sistema en sistemas en los que ya no es posible arrancar y/o iniciar sesión.

Ejemplos comunes son:

- reinstalar el gestor de arranque
- reconstrucción de la imagen **initramfs**

- actualizar o degradar paquetes
- restablecer una contraseña olvidada
- creación de software en un entorno root limpio

Capítulo 20: Gestores de paquetes

Sección 20.1: Cómo actualizar paquetes con el gestor de paquetes apt

La herramienta avanzada de paquetes (**A**dvanced **P**ackage **T**ool), llamada acertadamente gestor de paquetes **apt**, puede gestionar la instalación y eliminación de software en Debian, Slackware y otras distribuciones de Linux. A continuación, se muestran algunos ejemplos sencillos de uso:

update

Esta opción recupera y escanea los archivos Packages.gz, de forma que la información sobre paquetes nuevos y actualizados esté disponible. Para ello, introduzca el siguiente comando:

```
sudo apt update
```

o

```
sudo apt-get update
```

upgrade

Esta opción se utiliza para instalar las versiones más recientes de todos los paquetes instalados actualmente en el sistema. Los paquetes actualmente instalados con nuevas versiones disponibles se recuperan y actualizan; en ningún caso se eliminan los paquetes actualmente instalados, ni se recuperan e instalan los paquetes que aún no están instalados. Para actualizar, introduzca el siguiente comando:

```
sudo apt upgrade
```

o

```
sudo apt-get upgrade
```

dist-upgrade

Además de realizar la función de actualización, **dist-upgrade** también gestiona de forma inteligente el cambio de dependencias con nuevas versiones de paquetes. Si es necesario, intentará actualizar los paquetes más importantes a expensas de los menos importantes. Para ello, introduzca el siguiente comando:

```
sudo apt-get dist-upgrade
```

Sección 20.2: Cómo instalar un paquete con el gestor de paquetes pacman

Para buscar paquetes en la base de datos, buscando tanto en los nombres como en las descripciones de los paquetes:

```
pacman -Ss string1 string2 ...
```

Para instalar un solo paquete o una lista de paquetes (incluidas las dependencias), ejecute el siguiente comando:

```
sudo pacman -S package_name1 package_name2 ...
```

fuentes

Sección 20.3: Cómo actualizar paquetes con el gestor de paquetes pacman

Para actualizar un programa específico:

```
sudo pacman -S <programName>
```

Para actualizar todo el sistema:

```
sudo pacman -Syu
```

Sección 20.4: Cómo actualizar paquetes con yum

Yellowdog Updater, Modified, uno de los últimos vestigios de Yellow Dog Linux, es el gestor de paquetes utilizado por los sistemas Red Hat, Fedora y CentOS y sus derivados. Puede manejar la instalación y eliminación de software empaquetado como **rpms** para estas distribuciones de Linux. A continuación, se muestran algunos ejemplos sencillos de uso:

search

Este comando intentará localizar paquetes de software en los repositorios de software configurados que coincidan con los criterios de búsqueda dados, y mostrará el nombre / versión / ubicación del repositorio de las coincidencias que encuentre. Para utilizarlo, introduzca el siguiente comando:

```
yum search <queryString>
```

install

Este comando intentará localizar e instalar el software nombrado desde los repositorios de software configurados, localizando e instalando recursivamente también cualquier software prerequisite necesario. Para utilizarlo, introduzca el siguiente comando:

```
sudo yum install <packageName>
```

update

Esta opción se utiliza para instalar las versiones más recientes de todos los paquetes instalados actualmente en el sistema. Los paquetes actualmente instalados con nuevas versiones disponibles se recuperan y actualizan; también se recuperan e instalan nuevos prerequisites según sea necesario, y se eliminan los paquetes reemplazados u obsoletos. Para actualizar, introduzca el siguiente comando:

```
sudo yum update
```

A diferencia de **apt**, la mayoría de los comandos **yum** también comprobarán automáticamente si hay actualizaciones en los metadatos del repositorio si no se ha realizado una comprobación recientemente (o si se ha forzado a hacerlo) y recuperarán y escanearán los metadatos actualizados para que la información sobre paquetes nuevos y actualizados esté disponible antes de realizar la operación solicitada.

Capítulo 21: Compilación del kernel Linux

Sección 21.1: Compilación del núcleo Linux en Ubuntu

Atención: asegúrese de que dispone de al menos 15 GB de espacio libre en disco.

Compilación en Ubuntu >=13.04

Opción A) Utilizar Git

Usa git si quieres mantenerte sincronizado con el último código fuente del kernel de Ubuntu. Puede encontrar instrucciones detalladas en la Guía Git del núcleo. El repositorio git no incluye los archivos de control necesarios, por lo que debe construirlos por:

```
fakeroot debian/rules clean
```

Opción B) Descargar el archivo fuente

Descargar el archivo fuente - Esto es para los usuarios que quieren reconstruir los paquetes estándar de Ubuntu con parches adicionales. Utilice un comando follow para instalar las dependencias de compilación y extraer el código fuente (en el directorio actual):

1. Instale los siguientes paquetes:

```
sudo apt-get build-dep linux-image-`uname -r`
```

Opción C) Descargar el paquete fuente y compilar

Esto es para los usuarios que quieran modificar, o jugar, con el código fuente del kernel parcheado de Ubuntu.

1. Recupera la última fuente del kernel de kernel.org.
2. Extrae el archivo a un directorio y **cd** en él:

```
tar xf linux-*.tar.xz  
cd linux-*
```

3. Construye la interfaz de configuración de ncurses:

```
make menuconfig
```

4. Para aceptar la configuración predeterminada, pulse para resaltar `< Exit >` y, a continuación, .
5. Vuelva a pulsar Retorno para guardar la configuración.
6. Usa **make** para construir el kernel:

make

Tenga en cuenta que puede utilizar la bandera `-jem>` para compilar archivos en paralelo y aprovechar múltiples núcleos.

La imagen comprimida del kernel puede encontrarse en `arch/[arch]/boot/bzImage`, donde `[arch]` es igual a `uname -a`.

Créditos

Muchas gracias a todas las personas de Stack Overflow Documentation que ayudaron a proporcionar este contenido, más cambios pueden ser enviados a web@petercv.com para que el nuevo contenido sea publicado o actualizado.

Traductor al español

[rortegag](#)

Ztheo.tk	Capítulo 1
Aaron Skomra	Capítulo 16
Ajay Sangale	Capítulos 1 y 9
Anagh Hegde	Capítulos 4 y 14
Ani Menon	Capítulos 2 y 4
Arden Shackelford	Capítulo 12
Armali	Capítulo 1
Baard Kopperud	Capítulo 8
BrightOne	Capítulos 9, 13, 14 y 19
C.W.Holeman II	Capítulo 14
caped114	Capítulos 1 y 4
colemmonz	Capítulo 1
ctafur	Capítulo 17
DaveM	Capítulo 2
depperm	Capítulo 1
e.dan	Capítulo 1
embedded	Capítulo 8
Emmanuel Mathi	Capítulo 4
EsmaeelE	Capítulo 1
fdeslaur	Capítulo 3
Federico Ponzi	Capítulo 9
Filipe	Capítulo 14
Flamewires	Capítulo 10
FOP	Capítulo 12
foxtrot9	Capítulo 7
geek1011	Capítulo 11
Jarryd	Capítulo 1
Jensd	Capítulos 1 y 4
KerDam	Capítulo 1
Kiran Vemuri	Capítulos 6 y 17
kuldeep mishra	Capítulo 13
Léo Léopold Hertz 준영	Capítulo 21
lardenn	Capítulos 2 y 4
leeor	Capítulo 9
likewhoa	Capítulo 1
manav m	Capítulo 15
Manuel	Capítulo 14
Marsso	Capítulo 8
Mateusz Piotrowski	Capítulo 1
mattdm	Capítulos 2, 10 y 11
mertylldiran	Capítulo 5
Mike P	Capítulo 1
Mohammad	Capítulo 1
Nathan Osman	Capítulo 21
Naveen Chakravarthy	Capítulo 1

<u>Nikhil Raj</u>	Capítulo 2
<u>Not22</u>	Capítulo 8
<u>oznek</u>	Capítulo 4
<u>Paradox</u>	Capítulos 16 y 20
<u>parkydr</u>	Capítulo 7
<u>Philip Kirkbride</u>	Capítulos 4, 7, 12 y 20
<u>Quaker</u>	Capítulo 4
<u>Rajesh Rengaraj</u>	Capítulo 14
<u>Riley Guerin</u>	Capítulo 15
<u>Rubio</u>	Capítulos 1, 4 y 20
<u>S.Rohit</u>	Capítulos 3, 6 y 12
<u>Sava</u>	Capítulos 1, 4, 14 y 18
<u>Sergey Stolyarov</u>	Capítulo 2
<u>Sudip Bhandari</u>	Capítulo 1
<u>Teddy</u>	Capítulo 2
<u>Tejus Prasad</u>	Capítulo 1
<u>TiansHUo</u>	Capítulo 1
<u>Todd</u>	Capítulos 1 y 14
<u>vishram0709</u>	Capítulo 1
<u>Whoami</u>	Capítulo 1
<u>Y4Rv1K</u>	Capítulos 6 y 17
<u>Zumo de Vidrio</u>	Capítulo 1
<u>zyio</u>	Capítulo 5