



# **Preparación para el examen LPI 101**

## **Tema 103.3**

# **Administración de archivos**

## Créditos y licencia de uso

### Coordinación:

Manuel Guillán (xLekOx) [lpi@xleko.org](mailto:lpi@xleko.org)

### Traducción:

Miguel Castiella (ruton) [mcastiella@euskalnet.net](mailto:mcastiella@euskalnet.net)

Dani Donisa (kasei) [kasei@flashmail.com](mailto:kasei@flashmail.com)

Juan Maria Gil (Smooth) [yo@juanmaria.com](mailto:yo@juanmaria.com)

Manuel Guillán (xLekOx) [lpi@xleko.org](mailto:lpi@xleko.org)

Pablo Taboada (java) [ptaboada@wanadoo.es](mailto:ptaboada@wanadoo.es)

### Maquetación:

Manuel Guillán (xLekOx) [lpi@xleko.org](mailto:lpi@xleko.org)

Kiefer Von Jammo (Kiefer) [kiefer@khroon.net](mailto:kiefer@khroon.net)

Oscar Casal (ocs) [oscar@glug.es](mailto:oscar@glug.es)

Versión 1.2 (07-08-2004 3:00)

Distribuido por FreeUOC ([www.freeuoc.org](http://www.freeuoc.org)) bajo licencia: Attribution-NonCommercial-ShareAlike2.0 de commons creative



<http://creativecommons.org/licenses/by-nc-sa/2.0/>

## ÍNDICE

### Índice de contenido

#### Tema 103.3

|                                                             |    |
|-------------------------------------------------------------|----|
| Administración de archivos.....                             | 1  |
| Créditos y licencia de uso.....                             | 2  |
| ÍNDICE.....                                                 | 3  |
| Introducción.....                                           | 4  |
| Administrador de ficheros.....                              | 5  |
| Cambiando directorios.....                                  | 5  |
| Listando los contenidos de un directorio.....               | 7  |
| Determinando el tipo de fichero.....                        | 9  |
| Cambiando la fecha de acceso (time stamp).....              | 10 |
| Copiando Archivos.....                                      | 11 |
| Copiando y convirtiendo archivos con diferente formato..... | 12 |
| Moviendo Archivos.....                                      | 13 |
| Borrado de archivos.....                                    | 13 |
| Creando directorios.....                                    | 14 |
| Entendiendo la jerarquía del sistema de ficheros.....       | 15 |
| Localización de los directorios estándar.....               | 15 |
| Directorios del sistema.....                                | 16 |
| Localización de ficheros.....                               | 16 |
| PREGUNTAS PRE-TEST.....                                     | 17 |
| PREGUNTAS TEST.....                                         | 17 |
| RESPUESTAS PRE-TEST.....                                    | 18 |
| RESPUESTAS TEST.....                                        | 18 |
| Bibliografía y enlaces recomendados.....                    | 20 |

## **Introducción**

En este tema se tocarán temas referentes a la administración de los ficheros/directorios, se verá como copiar ficheros/directorios, moverlos, borrarlos, modificar sus propiedades, buscar archivos según su tamaño, tipo u hora.

Para terminar, se aprenderá como se usan los enlaces de ficheros y donde se encuentran generalmente muchos de los ficheros en un sistema Linux. El manejo de ficheros es la mayor parte del trabajo con sistemas Linux, por tanto es muy importante la comprensión de los apartados tratados a continuación.

Los comandos que se verán en este tema son:

- cp
- find
- mkdir
- mv
- ls
- rm
- rmdir
- touch

Así mismo, se harán ejercicios sobre los mismos al final del tema, que serán muy parecidos a los realizados en los exámenes.

Este tema tiene un peso (importancia) de 3 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

## Administrador de ficheros

Usar los comandos básicos de Unix para copiar y mover ficheros y directorios. Utilizar las opciones avanzadas del manejo de ficheros como la copia recursiva de múltiples ficheros y mover ficheros con un determinado patrón.

Cuando se trabaja con sistemas GNU/Linux, se necesita estar familiarizado con los comandos básicos del manejo de ficheros. En esta sección se cubrirán los más comunes, incluyendo los relacionados con el sistema de ficheros y los comandos para trabajar con directorios y ficheros.

Se necesitará saber para que se usa cada comando y las opciones más importantes de cara a la buena realización del examen.

Cuando se trabaja con ficheros en sistemas Linux, se ha tener en cuenta unas normas y restricciones. Los ficheros ocultos empiezan con un punto. Los nombres de ficheros pueden empezar con un número pero no pueden contener barras, signos de interrogación, asteriscos u otro tipo de caracteres reservados. La extensión de los ficheros no siempre es requerida, pero puede ser útil para un mantenimiento de los tipos de ficheros.

Las siguientes secciones examinan algunos de los comandos básicos cuando se trabaja con ficheros y directorios.

## Cambiando directorios

Los directorios en un sistema GNU/Linux se distribuyen en una estructura de árbol. El directorio /, conocido como directorio raíz, contiene un número de directorios del sistema. Cada uno de estos directorios puede contener subdirectorios, como corresponde a un sistema de árbol. La figura 3-1 muestra esta estructura de directorios. El directorio del sistema y su localización son tratados en este capítulo.

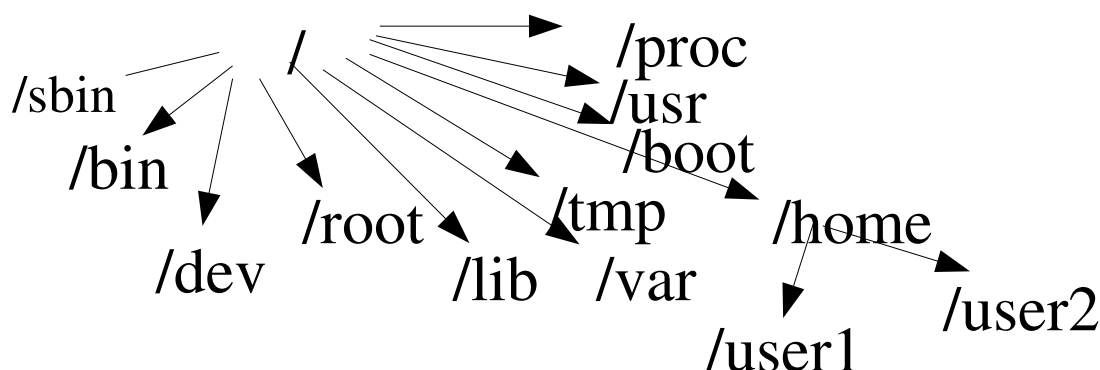


Figura 3-1: Estructura de directorios básica de Linux



El comando `cd` se utiliza para moverse por los directorios en un shell de Linux; `cd` viene de *change directory*. Este es uno de los comandos más simples usados en Linux. El comando `pwd` (*present working directory*) nos da el directorio actual, en el que se está en el momento de ejecutar el comando. Cuando se usa sin parámetros, el comando `cd` cambia del directorio de trabajo al directorio `home`. Es posible especificar también el directorio que quieres que sea tu directorio de

trabajo.

Ésta es la sintaxis del comando cd:

```
$ cd /directory
```

El comando cd se puede usar con el slash (/) cuando se quiere especificar un directorio absoluto, empezando desde el directorio raíz. Ésto permite moverse a cualquier parte especificando el directorio absoluto.

Sin el slash, cd buscará el directorio a partir de donde se ejecuta el comando (path relativo). A continuación se muestra un ejemplo del path relativo.

```
$ pwd
/home
$ cd angie
$ pwd
/home/angie
```



El comando cd se puede usar también con variables de entorno del sistema.

Los directorios “.” y el “..” son ficheros ocultos que existen en cada directorio. El “.” representa el directorio actual mientras que el “..” representa el directorio padre. Para el directorio / el “..” apunta a él mismo.

Otro símbolo utilizado es el carácter ~. Este carácter se usa para representar el directorio home del usuario.

Para terminar, el carácter – se puede usar para volver al directorio anterior.

Otro carácter especial que puede usarse con el comando cd es el \*. Este carácter es usado para representar todos los caracteres.

El siguiente comando lista todos los ficheros de un directorio, que empiecen por la letra a, incluyendo los que solo sean una a.

```
$ ls a*
```

El siguiente ejemplo retrocede una vez en la estructura de directorios.

```
$ cd ..
```

**Nota.** Cuidado con el espacio entre el comando y los puntos. Este espacio es requerido en Linux, a no ser que creamos un alias para cd..

El siguiente es un ejemplo del uso de variables de entorno con el comando cd para cambiar del directorio donde estamos al directorio home del usuario. El símbolo \$ se usa para especificar una variable de entorno en el comando.

```
$ cd $HOME
```

El comando cd no tiene opciones. Éste comando es uno de los más utilizados en Linux.

## Listando los contenidos de un directorio



Otro comando muy utilizado en Linux es el comando `ls`, que se utiliza para ver el contenido de un directorio. Cuando el comando es utilizado sin opciones ni parámetros, lista todos los ficheros y directorios localizados en el directorio de trabajo que no tienen marca de archivos ocultos. Por defecto, se muestra en orden alfabético:

```
$ ls
```

El comando `ls` acepta paths como argumentos y entonces mostrará los ficheros y directorios de ese path. Se pueden usar paths absolutos:

```
$ ls /home/angie/stuff
```

También se pueden usar paths relativos:

```
$ ls stuff
```

El argumento puede contener cadenas de caracteres o caracteres especiales, en cuyo caso se mostrará un listado de todos los archivos y directorios que cumplan la condición.

```
$ ls a*  
abc123names abcnames alphanames alteredservices
```

Cuando se usa un path en conjunción con caracteres especiales, se muestra el path con el resultado:

```
$ ls /home/angie/stuff/n*  
/home/angie/stuff/namelist /home/angie/stuff/nicks
```

```
$ ls stuff/n*  
stuff/namelist /stuff/nicks
```

El comando `ls` pueden aceptar opciones como argumento. Hay un gran número de opciones para éste comando que permiten un gran control sobre el resultado. En la tabla siguiente se muestran las opciones más usadas:

Tabla 3-1: Opciones del comando `ls`

| Opción | Uso                                                              |
|--------|------------------------------------------------------------------|
| -a     | Lista todos los contenidos del directorio.                       |
| -A     | Trabaja como el -a excepto que no lista "." y el ".."            |
| -B     | No se lista los ficheros que finalizan con ~                     |
| -d     | Muestra el nombre del directorio en el listado                   |
| -L     | Muestra la información para los ficheros enlaces o referenciales |
| -R     | Muestra los directorios recursivamente.                          |

Una variedad de opciones pueden ser usadas para especificar la información mostrada en un listado de ficheros. La tabla 3-2 cubre la información de las opciones de listado con el comando `ls`.

Tabla 3-2 Listado de las opciones de información de ls

| Opción | Uso                                                                                                                                     |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------|
| -G     | Especifica que grupo de información no se muestra                                                                                       |
| -I     | Muestra el número inode                                                                                                                 |
| -l     | Muestra el tipo de fichero, permisos, contador de enlaces permanentes, propietario, grupo propietario y fecha de la última modificación |
| -o     | Muestra la misma información que -l menos la información de grupo que es excluida                                                       |
| -s     | Muestra el tamaño del fichero en bloques de 1024 Kb                                                                                     |

El comando ls utiliza a veces opciones para ordenar la salida, las cuales se muestran en la tabla 3-3

Tabla 3-3 Listado de las opciones de ordenación de la salida de ls

| Opción | Uso                                                                                          |
|--------|----------------------------------------------------------------------------------------------|
| -c     | Muestra el resultado acorde con la fecha de modificación o la fecha de modificación de inode |
| -f     | Muestra el resultado con el orden en que han sido salvados en el directorio                  |
| -r     | Muestra el listado en orden inverso                                                          |
| -S     | Muestra el listado de acuerdo al tamaño, del más grande al más pequeño                       |
| -t     | Muestra el listado de acuerdo a la fecha de modificación, mostrando primero el más reciente. |
| -u     | Muestra el listado de acuerdo al último acceso, empezando por el más reciente.               |

La salida producida por el comando ls puede ser también controlada con otra serie de opciones. Estas opciones se muestran en la tabla 3-4.

Tabla 3-4 Listado de las opciones de apariencia de la salida de ls

| Opción  | Uso                                                                                                                                                                                                                                          |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -l      | La salida se muestra una fila por línea                                                                                                                                                                                                      |
| -C      | La salida se muestra en columnas                                                                                                                                                                                                             |
| -F      | Muestra el nombre del fichero con una letra para especificar el tipo de fichero                                                                                                                                                              |
| -k      | Muestra el tamaño del fichero en Kb                                                                                                                                                                                                          |
| -m      | Muestra los nombres de ficheros separados por comas                                                                                                                                                                                          |
| -n      | Muestra el usuario y el número de grupo                                                                                                                                                                                                      |
| -p      | Muestra los nombres de ficheros con un carácter para especificar el tipo                                                                                                                                                                     |
| -x      | Muestra el nombre de fichero en columnas ordenadas horizontalmente                                                                                                                                                                           |
| -T COLS | Asume que cada parada de tabulación está a cols columnas de ancho; el valor predeterminado es 8. ls emplea tabuladores donde es posible en la salida, por eficiencia. Si cols es cero, no usa tabuladores para nada.                         |
| -W COLS | Asume que la pantalla tiene cols columnas de ancho. El valor predeterminado se toma del controlador de terminal si es posible; si no, se emplea la variable de ambiente COLUMNS si está definida; de otro modo el valor predeterminado es 80 |

Todas éstas opciones se pueden combinar para crear una salida muy específica usando el comando ls. Los siguientes son unos ejemplos de como éstas opciones se pueden usar para controlar que



datos se muestran y como se muestran. A través de las tablas anteriores se puede comprobar que opciones se usan.



Saber usar el comando ls, junto a sus posibles opciones, es muy importante de cara al examen.

El siguiente ejemplo lista los nombres de ficheros de acuerdo con la fecha del último acceso, con la / para mostrar los directorios:

```
$ ls -pu  
morestuff/  nicks list  readmes  alphanames  namelist
```

El siguiente ejemplo muestra el contenido del directorio /home/angie/stuff y todos sus subdirectorios:

```
$ ls -R /home/angie/stuff
```

A través de los ejemplos mostrados, se puede observar que el comando ls puede ser una potente herramienta para sacar información de los ficheros y directorios. El comando se usa muchas veces en conjunción con otros comandos y ficheros usando filtros y redireccionamientos. Ésto permite salvar la salida a un fichero o procesarlo con otras utilidades.

### Determinando el tipo de fichero



El comando ls provee de mucha información cuando se examinan ficheros, pero no muestra información sobre el tipo de contenido de los mismos.

El comando file puede ser usado para aprender más sobre el tipo contenido de los ficheros en un sistema Linux. La salida del comando file incluye una de las siguientes palabras: text, executable, data o directory. Este comando acepta argumentos para especificar que ficheros examina. Se pueden usar una serie de opciones con este comando, como se muestran en la tabla 3-5

Tabla 3-5 Opciones usadas con file

| Opción             | Uso                                                                                                                                              |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| -b                 | Especifica que el nombre de fichero no se muestre en la salida.                                                                                  |
| -f <nombrefichero> | Especifica que el fichero <nombrefichero> contiene los nombres de los ficheros a examinar.                                                       |
| -n                 | Muestra el resultado después del chequeo. Esto puede ser útil cuando trabajamos con una serie de ficheros que van a ser enviados a otro comando. |
| -v                 | Muestra la versión                                                                                                                               |
| -z                 | Intenta examinar el contenido de los ficheros comprimidos.                                                                                       |

Algunos ejemplos del comando file y sus opciones.

En el primer ejemplo, el \* hace que se muestre el tipo de contenido de todos los archivos del directorio actual.

```
$ file *
```

En el siguiente ejemplo, el nombre de fichero no es incluido dentro del listado.

```
$ file -b /etc/lilo.conf
```

En el ejemplo siguiente, el comando file se usa para examinar el tipo de contenido de un fichero comprimido:

```
$ file -z /usr/info/tar.info.gz
```

## Cambiando la fecha de acceso (time stamp)



El comando touch permite cambiar la fecha de último acceso de un fichero. Si se especifica el nombre de fichero, pero el fichero no existe, se crea un fichero vacío con ese nombre. Las opciones del comando touch se pueden ver en la tabla 3-6.

Tabla 3-6 Opciones usadas con touch

| Opción      | Uso                                                                                 |
|-------------|-------------------------------------------------------------------------------------|
| -a          | Solo cambia la fecha de acceso en el fichero                                        |
| -c          | No crea un fichero si el nombre no existe.                                          |
| -d <string> | Usado para especificar el formato de la fecha en el fichero usando la fecha actual. |
| -r <file>   | Utiliza la fecha de <file> en vez de la del sistema                                 |
| -t <TIME>   | Utiliza una fecha como argumento en vez de la del sistema.                          |

El comando touch permite usar la fecha actual por defecto o especificar la fecha de último acceso. El comando se usa a menudo para crear ficheros vacíos con un nombre específico. Esto puede ser útil cuando se ejecuta un script que necesita un fichero que no existe.

El sistema de rotado de los ficheros de log, es un ejemplo de uso del comando touch. En dicho script, se renombra el fichero de log actual a una versión más antigua. Tras el renombrado, con el comando touch, se crea el fichero que seguirá almacenando los logs:

```
...
mv syslog syslog.1      -> renombramos el fichero
touch syslog            -> se crea de nuevo el fichero, vacío
...
```

El siguiente ejemplo muestra un script en el cual el comando touch se usa para crear un fichero vacío.

```
$ cp mylogs stuff/mylogs.`date +%m.%d.%y`
$ rm -f mylogs
$ touch mylogs
```

Copia el contenido de mylogs al directorio stuff con el nombre mylog.mes.dia.año (por ejemplo mylogs.12.11.00) El fichero original mylog es borrado y entonces se usa el comando touch para crearlo de nuevo vacío.

Uno de los usos habituales del comando touch, es la creación de archivos vacíos.

## Copiando Archivos



Cuando se trabaja con archivos en cualquier sistema, generalmente se necesita copiar archivos. Linux incluye dos comandos para copiar archivos.

El comando `cp` se usa para copiar archivos y directorios. Es el comando estándar para copiar archivos de una localización del sistema a otra.

Cuando se quiere copiar archivos de un formato de archivo a otro, se hará uso el comando `dd`.

El comando `cp` (abreviatura de copy) se utiliza para la copia estándar de archivos en sistemas Linux. Este comando se usa para crear una copia nueva e independiente del archivo o directorio original. Se pueden usar muchas opciones con el comando `cp` para modificar las copias creadas. Las opciones se resumen en la tabla 3-7:

Tabla 3-7 Opciones del comando `cp`

| Opción | Uso                                                                                                                                                                      |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a     | Especifica que los enlaces y atributos del archivo original deben ser transferidos a la nueva copia.                                                                     |
| -d     | Especifica que los enlaces se deben mantener cuando se copia.                                                                                                            |
| -f     | Sobreescribe cualquier archivo de destino existente.                                                                                                                     |
| -i     | Pregunta antes de sobreescribir cualquier archivo de destino existente.                                                                                                  |
| -l     | Especifica que se creen enlaces fuertes (ver más adelante en este capítulo) en lugar de copias del archivo.                                                              |
| -p     | Mantiene el propietario, grupo, permisos y timestamp del archivo original.                                                                                               |
| -r     | Copia directorios y sus contenidos de forma recursiva mientras copia cada archivo como archivo estándar. Esta opción no podrá ser usada con algunos archivos especiales. |
| -R     | Copia directorios y sus contenidos de forma recursiva, manteniendo los directorios.                                                                                      |
| -s     | Crea enlaces simbólicos (ver más adelante) de los archivos que no sean directorios.                                                                                      |
| -v     | Muestra todos los nombres de los archivos según se van copiando.                                                                                                         |

Estas opciones se pueden combinar cuando se copian archivos. Junto con las opciones, también se usan argumentos con el comando `cp`. La sintaxis correcta del comando es la siguiente:

```
cp -opciones origen destino
```

Cuando el destino especificado es un directorio, el archivo de origen se copia a ese directorio con el mismo nombre que el archivo original. Cuando el destino especificado no es un directorio, el archivo de origen se copia a la localización especificada con el nombre del destino.

Los siguientes son algunos ejemplos de uso del comando `cp`.

En el primer ejemplo, el archivo “marital” se copia al directorio “stuff”.

```
$ cp marital stuff
```

En el segundo ejemplo, el directorio “/home/angie/stuff” y su contenido se copia al directorio “/home/angie/otrostuff”. Esto es particularmente útil, ya que permite copiar el contenido entero de un directorio usando un único comando:

```
$ cp -r /home/angie/stuff /home/angie/otrostuff
```



**Pregunta de Examen:** Ésta es una opción importante que aparece frecuentemente como pregunta de examen.

## Copiando y convirtiendo archivos con diferente formato



El comando dd (abreviatura de Direct Dump) se usa para copiar y convertir archivos de un formato a otro simultáneamente. Este comando tiene diferentes opciones y diferente sintaxis que el comando cp. La sintaxis utilizada para el comando dd es la siguiente:

```
dd [opciones]
```

El comando dd, por defecto, escribe datos desde la entrada estándar hacia la salida estándar. Las opciones se utilizan para cambiar estos valores por defecto. Las opciones para el comando dd se muestran en la tabla 3-8:

Tabla 3-8 Opciones del comando dd

| Opción       | Uso                                                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------|
| if=FILE      | Especifica la localización del archivo de origen para ser utilizado en lugar de la entrada estándar. Éste es el archivo de origen.  |
| of=FILE      | Especifica la localización del archivo de destino para ser utilizado en lugar de la salida estándar. Éste es el archivo de destino. |
| ibs=BYTES    | Especifica el número de bytes leídos de cada vez.                                                                                   |
| obs=BYTES    | Especifica el número de bytes escritos de cada vez.                                                                                 |
| bs=BYTES     | Especifica el número de bytes a escribir y leer de cada vez.                                                                        |
| cbs=BYTES    | Especifica el número de bytes a convertir de cada vez.                                                                              |
| skips=BLOCKS | Especifica los bloques a saltar en el archivo de origen antes de empezar a copiar.                                                  |
| seek=BLOCKS  | Especifica los bloques a saltar en el archivo de destino antes de empezar a escribir.                                               |
| count=BLOCKS | Especifica los bloques a copiar del archivo de origen en lugar de copiar el archivo completo.                                       |

El comando dd se puede utilizar para variedad de tareas especiales. Por ejemplo:

```
$ /bin/dd if=/dev/hda5 bs=512 count=1 of=/mnt/win_c/bootsek.lin
```

Esto creará un archivo llamado bootsek.lin en la partición montada en /mnt. El archivo se escribirá como un bloque con un tamaño de 512 bytes.

En este ejemplo, el comando dd se utiliza para copiar un archivo desde una partición a otra partición

formateada con el sistema de archivos FAT. Este comando es útil también para trabajar con sistemas de copia de seguridad de tipo cinta.

## Moviendo Archivos



Se puede mover un archivo manualmente copiándolo a la nueva localización y borrando luego el archivo original. Sin embargo, Linux incluye un comando para mover archivos que automatiza esta tarea. El comando `mv` (abreviatura de `move`) permite mover y renombrar archivos en sistemas Linux. Este comando funciona como el comando `cp`, utilizando la misma sintaxis. Las opciones para el comando `mv` son algo diferentes y se resumen en la tabla 3-9:

Tabla 3-9 - Opciones del comando `mv`

| Opción          | Uso                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------|
| <code>-f</code> | Borra los archivos existentes sin pedir confirmación.                                                         |
| <code>-i</code> | Pide confirmación al usuario antes de sobrescribir archivos.                                                  |
| <code>-u</code> | Especifica que los archivos no serán movidos al destino si tienen fecha de modificación igual o más reciente. |
| <code>-v</code> | Muestra por pantalla los archivos movidos.                                                                    |

En el siguiente ejemplo, todos los archivos que cumplan el patrón, empezando por “archivo” serán movidos al directorio “misarchivos” y mostrará por pantalla los nombres de los archivos movidos:

```
$ mv -v archivo* misarchivos
```

En el segundo ejemplo, el directorio “misarchivos” se renombra a “archivos”:

```
$ mv misarchivos archivos
```

## Borrado de archivos



Otra tarea frecuentemente necesaria cuando trabajamos con archivos y directorios es la eliminación de los mismos. El comando `rm` (abreviatura de `remove`) se utiliza para borrar archivos y directorios en sistemas Linux. Utiliza la siguiente sintaxis:

```
rm -opciones ARCHIVO
```

Se pueden utilizar muchas opciones con el comando `rm`. Las utilizadas más frecuentemente son las que se muestran en la tabla 3-10:

Tabla 3-10 - Opciones del comando rm

| Opción | Uso                                                                                          |
|--------|----------------------------------------------------------------------------------------------|
| -d     | Utilizada por el superusuario. Elimina directorios sin tener en cuenta si están vacíos.      |
| -f     | Ejecuta el comando sin pedir confirmación, incluso si los archivos especificados no existen. |
| -i     | Pide confirmación al usuario para eliminar los archivos.                                     |
| -r     | Elimina el contenido del directorio de forma recursiva.                                      |
| -v     | Elimina archivos y muestra por pantalla los nombres de los archivos eliminados.              |

En el siguiente ejemplo se borrarán todos los archivos del directorio actual (pwd) cuyo nombre comience con “nn”:

```
$ rm nn*
```

El segundo ejemplo eliminará todos los archivos del directorio “archivos” de forma recursiva, así como el propio directorio y muestra por pantalla los nombres de los archivos eliminados:

```
$ rm -frv archivos
removing archivos/archivo1
removing archivos/archivo2
removing archivos/archivo3
removing archivos/archivo4
removing archivos/archivo5
removing the directory itself: archivos
```

### Creando directorios



En todo este apartado se ha tratado el tema de la creación y eliminación de archivos, así como de la eliminación de directorios. Ahora trabajaremos con el comando *mkdir* (abreviatura de make directory), que se utiliza para crear directorios. Es un comando muy simple, que creará un directorio cada vez. Cuando se utiliza con la opción *-p*, se crearán también los directorios padres si es necesario.

Para crear los directorios:

```
/home/angie/nuestrosarchivos,  
/home/angie/nuestrosarchivos/misarchivos y  
/home/angie/nuestrosarchivos/tusarchivos
```

utilizaremos el siguiente comando:

```
$ mkdir /home/angie/nuestrosarchivos \  
/home/angie/nuestrosarchivos/misarchivos \  
/home/angie/nuestrosarchivos/tusarchivos
```

En el comando escrito, el directorio “nuestrosarchivos” se crea primero. Entonces el sistema crea los directorios “misarchivos” y “tusarchivos” sin el directorio “nuestrosarchivos”.

## Entendiendo la jerarquía del sistema de ficheros

El sistema de ficheros de Linux sigue una estructura en forma de árbol. La raíz (/) contiene los directorios principales. Cada uno de los directorios contenidos en el directorio raíz puede, a su vez, contener más directorios. Es importante familiarizarse con algunos directorios clave del sistema de ficheros, así como entender qué función llevan a cabo. En esta sección nos centraremos en algunos de estos directorios clave.

## Localización de los directorios estándar



En los siguientes directorios se encuentran algunos de los ficheros estándar de un sistema Linux:

-/etc : Contiene muchos de los ficheros de configuración y guiones (scripts) del sistema, así como ficheros de configuración propios de algunos programas.

-/etc/skel : Contiene los ficheros que se copian en el directorio del usuario cuando éste es creado.

-/usr : Contiene los subdirectorios donde se almacenan las aplicaciones y ficheros fuente usados por los usuarios, así como también librerías usadas por los programas. Hay varios de estos subdirectorios destacables:

-/usr/bin : Aquí viven los programas de las diversas aplicaciones del sistema. Cada usuario debería tener este subdirectorio en su variable de PATH.

-/usr/sbin : Aquí viven los programas usados por el administrador (o superusuario).

-/usr/local : Aquí se instalan por defecto las aplicaciones que no forman parte del sistema operativo. En concreto cabe destacar las siguientes:

-/usr/local/bin : Contiene los ejecutables de dichas aplicaciones.

-/usr/local/sbin : Contiene los ejecutables de dichas aplicaciones que solo puede ejecutar el administrador.

-/var/log : Contiene los ficheros de traza o log del sistema.

-/var/spool : Contiene la bandeja de correo y la cola de impresión, con los correspondientes ficheros de correo electrónico de cada usuario además de los ficheros que hubieran mandado a imprimir.

-/bin : Aquí viven los programas ejecutables más importantes del sistema operativo. También se encuentran los binarios usados durante el inicio del sistema.

-/sbin : Contiene los programas ejecutables usados por el administrador.

## Directorios del sistema

Algunos de los directorios con propósitos específicos que deberíamos conocer son los siguientes:

- / : Es el directorio raíz (de root, en inglés). De él cuelgan el resto de subdirectorios del sistema.
- /root : Es el directorio personal del administrador.
- /home : Es el directorio que contiene los directorios personales de los usuarios.
- /boot : Contiene los ficheros usados por el kernel durante el arranque.
- /dev : Contiene los ficheros de acceso a los dispositivos configurados en el sistema.
- /proc : proc es un directorio virtual (esto significa que sólo existe en tiempo de ejecución) usado por el núcleo del sistema (kernel) para comunicarse y almacenar ficheros con información del sistema.

## Localización de ficheros

Tarde o temprano va a ser necesario encontrar algún fichero o programa. Para ello Linux proporciona unas cuantas herramientas:

**find**

Se usa para buscar ficheros. Find empieza en un directorio y luego desciende por sus subdirectorios buscando el patrón indicado. Si no se especifica ningún directorio de búsqueda, *find* recoge el valor de la variable PWD, es decir, asume el directorio actual. Su sintaxis es como sigue:

```
find /ruta expresión
```

**locate**

Locate es una forma segura de indexar ficheros en el sistema para poder buscarlos más rápidamente en un futuro. Para ello guarda rutas -además de permisos y características de pertenencia, para preservar la privacidad- en una base de datos llamada slocate. Su sintaxis es como sigue:

```
locate -opciones argumentos
```

**which**

El comando *which* devolverá por la salida estándar aquello que ejecutaría la shell (dónde se encuentra el comando). Devolverá el comando con la ruta completa. La búsqueda la realiza en las rutas encontradas en el PATH. El comando *which* es útil para verificar que el comando que se quiere ejecutar es realmente el que se desea. Su sintaxis es como sigue:

```
which -opciones argumentos
```

**whereis**

Busca ficheros fuente, binarios y secciones de páginas de manual para los comandos especificados como argumentos. A esos argumentos se les quita los segmentos de ruta que pudieran llevar, los prefijos y las extensiones. Las búsquedas se realizan en una lista de rutas estándar.



## PREGUNTAS PRE-TEST

1. ¿Que comando de búsqueda de ficheros utiliza una base de datos?
2. ¿Donde se encuentra el núcleo (kernel) del sistema?
3. ¿Que tipo de link puede apuntar a distintos sistemas de ficheros?
4. ¿Que comando se utiliza para listar los ficheros de un directorio?
5. ¿Que comando se utiliza para crear un directorio?
6. ¿Que comando puede utilizarse para crear un fichero vacío?
7. ¿Que comando se emplea para buscar un comando en la ruta de PATH?
8. ¿Que comando busca recursivamente un fichero en un directorio?

## PREGUNTAS TEST

1. ¿Cual de los siguientes ejemplos muestra la utilización de un path absoluto?
  - A. ls
  - B. ls -al
  - C. ls /home/angie
  - D. /bin/ls
2. El comando \_\_\_\_\_ mostrará todas las entradas que comiencen por la letra a en el directorio actual.
3. ¿Cual de los siguientes comandos creará un nuevo fichero con el nombre nuestrogrupo?
  - A. file nuestrogrupo
  - B. touch nuestrogrupo
  - C. ls nuestrogrupo
  - D. mkfile nuestrogrupo
4. ¿Que comando se utiliza para cambiar el nombre del fichero nuestrogrupo a miggrupo?
  - A. rn nuestrogrupo miggrupo
  - B. rn miggrupo nuestrogrupo
  - C. mv nuestrogrupo miggrupo
  - D. mv miggrupo nuestrogrupo
5. El comando \_\_\_\_\_ se utiliza para copiar ficheros con conversión simultánea.
6. ¿Que directorio contiene el núcleo (kernel) del sistema?
  - A. /etc
  - B. /
  - C. /boot
  - D. /proc
7. ¿Que directorio contiene los ficheros de configuración del sistema?
  - A. /etc
  - B. /
  - C. /boot
  - D. /proc

8. ¿Que directorio contiene el directorio home del usuario root?
- A. /home
  - B. /root
  - C. /sbin
  - D. /usr
9. ¿Que directorio contiene los ficheros de correo?
- A. /proc/mail
  - B. /var/spool
  - C. /var/mail
  - D. /usr/mail
10. ¿Que utilidad se utiliza para actualizar la base de datos slocate?
- A. locate
  - B. find
  - C. whereis
  - D. updatedb
11. El comando \_\_\_\_\_ se utiliza para mostrar el path de un comando determinado.
12. ¿Cual de los resultados siguientes ocurre cuando se copia un soft link?
- A. Se crea una nueva copia del soft link.
  - B. Se crea un hard link hacia el fichero original.
  - C. Se crea un hard link hacia el soft link.
  - D. Se crea una nueva copia del fichero original.

### RESPUESTAS PRE-TEST

1. El comando locate utiliza la base de datos slocate para encontrar los ficheros.
2. El kernel del sistema se encuentra en el directorio /boot.
3. Los soft links pueden apuntar a diferentes sistemas de ficheros.
4. El comando ls se utiliza para listar los ficheros de un directorio.
5. El comando mkdir se utiliza para crear un directorio.
6. El comando touch puede utilizarse para crear un fichero vacío.
7. El comando whereis se utiliza para buscar un comando en la declaración PATH.
8. El comando find busca recursivamente un fichero en un directorio.

### RESPUESTAS TEST

1. **D.** El path absoluto al comando ls es /bin/ls. Para más información mira la sección “Manejando ficheros”
2. **ls a\***. El comodín se utiliza para especificar cualquier carácter o conjunto de caracteres. Este ejemplo listaría los ficheros y directorios llamados 'a' o cuyo nombre comience por 'a'. Para más información mira la sección “Listando el contenido del directorio”.
3. **B.** Pueden crearse ficheros con el comando touch si el nombre de fichero utilizado no existiese. Si existiese, simplemente se cambiaría la hora de acceso.

### *Tema 103.3 Administración de archivos*

Para más información mira la sección “Cambiando marca de hora en ficheros”.

**4. C.** El nombre y la ubicación de los ficheros se cambia con el comando mv. Para más información mira la sección “Moviendo ficheros”.

**5. dd.** El comando dd, o volcado de directorio, se utiliza para convertir y copiar ficheros. Para más información mira la sección “dd”.

**6. C.** El kernel del sistema se encuentra en el directorio /boot. Para más información mira la sección “Directorios del sistema”.

**7. A.** Los ficheros de configuración del sistema se encuentran en el directorio /etc. Para más información mira la sección “Ubicación de los ficheros estándar”.

**8. B.** El directorio home del usuario root es /root. Para más información mira la sección “Directorios del sistema”.

**9. B.** Los ficheros de correo se encuentran en el directorio /var/spool. Para más información mira la sección “Directorios del sistema”.

**10. D.** La base de datos slocate se actualiza con el comando updatedb. Para más información mira la sección “locate”.

**11. which.** El comando which muestra el path del comando especificado. Para más información mira la sección “which”.

**12. D.** Cuando se copia un soft link, se crea una nueva copia del fichero original en el destino especificado. Para más información mira la sección “Enlaces simbólicos”.

## **Bibliografía y enlaces recomendados**

LPIC 1 Certification Bible (Bible) by Angie Nash, Jason Nash  
John Wiley & Sons; Bk&CD-Rom edition (July 1, 2001) ISBN: 0764547720

LPI Linux Certification in a Nutshell by Jeffrey Dean  
O'Reilly & Associates; 1st ed edition (May 15, 2001) ISBN: 1565927486

CramSession's LPI General Linux Part 1 : Certification Study Guide  
CramSession.com; ISBN: B000079Y0V; (August 17, 2000)

Referencias Unix Reviews

<http://www.unixreview.com/documents/s=7459/uni1038932969999/>

Página LPI: [www.lpi.org](http://www.lpi.org)

Apuntes IBM: <http://www-106.ibm.com/developerworks/edu/l-dw-linux-lpir21-i.html>

Manuales GPL: <http://www.nongnu.org/lpi-manuals/>