

7.- Trabajar con la shell

7.1.- La necesidad de la línea de comandos

Si bien es verdad que el entorno gráfico de usuario para Linux ha ganado popularidad en los últimos tiempos, también es cierto que no todas las necesidades diarias pueden ser cubiertas con unos cuantos clics de ratón. En estos casos la línea de comandos nos ofrece una gran flexibilidad y eficiencia sobre todo, en el caso de los computadores Linux más antiguos, que no disponen de recursos suficientes para satisfacer las necesidades de hardware de algunos entornos gráficos, resultando muy importante, poder trabajar mediante programas basados en texto.

En modo texto están disponibles seis consolas virtuales entre las que puede navegar utilizando las combinaciones de teclas **Alt + F1** a **Alt + F6**. La séptima consola está reservada para **X11**.

7.2.- Introducción al Bash

Si observamos la barra de tareas encontraremos el icono de un monitor con una concha shell. Al pulsar sobre este icono se abrirá la ventana de la consola en la que puede introducir comandos. La ventana que aparece es un intérprete de comandos; normalmente se trata del intérprete **Bash** (Bourne again shell).

El Bash fue desarrollado en el marco del proyecto GNU y es, sin duda, el derivado más extendido de la Bourne Shell (sh). Cuando abra el Bash, verá sobre la primera línea el llamado Prompt (indicador de mandatos) que normalmente se compone del nombre de usuario, del nombre de la computadora, así como de la localización actual en el árbol de directorios (path). También puede ser configurado de forma individual. Cuando el cursor se encuentre detrás de este prompt, podrá introducir comandos directamente.

Comandos

Los comandos se componen de distintos elementos. En primer lugar se debe introducir siempre la palabra de comando y seguidamente los parámetros u

opciones. Los comandos se ejecutan sólo cuando se presiona **Return**.

Hasta ese momento podrá editar la línea de comandos sin ningún problema, añadir cualquier opción o corregir fallos.

Uno de los comandos más utilizados es **ls**, que se puede usar sólo o con los llamados argumentos. Si sólo introduce **ls** en la consola, se le mostrará el contenido del directorio en el que se encuentra en ese momento. Las opciones van precedidas de un guión.

Si introduce el comando **ls -l**, podrá ver el contenido del mismo directorio, sólo que de forma detallada. Verá, además del nombre del archivo, la fecha en que se creó, el tamaño del archivo en bytes y otras características que se explicarán más adelante.

Una de las opciones más importantes que también están disponible para muchos otros comandos es **--help**. Si por ejemplo introduce **ls --help**, se visualizarán todas las opciones para el comando **ls**.

Con el comando **ls** puede ver también el contenido de otros directorios que no sean el actual. Introduzca, por ejemplo, **ls -l Desktop** y seguidamente verá el contenido del subdirectorio Desktop.

Archivos y directorios

Para trabajar de forma eficiente con la shell, debe conocer la estructura de archivos y de directorios en Linux. Los directorios son carpetas electrónicas en las cuales se pueden depositar archivos o programas tal como subdirectorios. El directorio de mayor jerarquía es el directorio raíz, que se representa por **/**. Desde este directorio podremos acceder a todos los demás.

El directorio **/home** como ya sabemos contiene los directorios personales en los que todos los usuarios dejan sus archivos.

Funciones del Bash

Dos funciones importantes de la shell le simplificarán enormemente su trabajo:

⌘ **El historial:** Si desea volver a ejecutar una orden que ya haya introducido previamente, basta con pulsar la tecla “flecha arriba” hasta que aparezca el comando deseado. Si el comando que busca se encuentra delante de su posición actual pulse la tecla “flecha abajo”. También puede editar la línea de comando si desea modificar la orden. Desplácese con ayuda de las teclas del cursor hasta el sitio deseado y corrija el error.

⌘ **La función de expansión:** Esta función completa el nombre de un archivo del cual sólo se han introducido las primeras letras, en caso de que éste pueda ser identificado de forma inequívoca. Para ello pulse la tecla **Tab**. En caso de que haya más de un archivo que empiece con esas letras, obtendrá una lista de todos estos archivos pulsando dos veces seguidas la tecla **Tab**.

Ejemplo: Trabajar con archivos

Ahora que ya sabemos cómo se estructura un comando, qué directorios existen en Linux y cómo puede simplificar un poco el trabajo con el bash, haremos un pequeño ejercicio para que pueda poner en práctica estos conocimientos:

- 1 Abrir una consola haciendo clic sobre el icono del monitor y la concha.
- 2 Introduzca el comando **ls**. Verá el contenido de su directorio privado.
- 3 Cree un nuevo subdirectorio con **mkdir** make directory de nombre prueba. Para ello introduzca: **mkdir prueba** (puede escribir **md** en vez de **mkdir**).
- 4 Inicie el editor KWrite pulsando las teclas **Alt + F2** e introduciendo kwrite en la casilla de texto, lo que provocará que se abra una nueva ventana. Escriba un par de letras y grabe el archivo con el nombre Archivoprueba con una A mayúscula, puesto que Linux diferencia entre mayúsculas y minúsculas.
- 5 Ahora mire nuevamente el contenido de su directorio. Pero esta vez, en

lugar de volver a introducir el comando **ls**, presione dos veces seguidas sobre la tecla “flecha arriba”. A continuación volverá a ver **ls** en la línea de comandos, y sólo tendrá que pulsar sobre la tecla **Return** para volver a ejecutarlo. Ahora verá el nuevo subdirectorio prueba en color azul y el Archivoprueba en color negro, puesto que los directorios siempre vienen indicados en azul y los archivos en negro.

6 Ahora con el comando **mv** move, podemos mover el Archivoprueba al subdirectorio prueba. Para facilitar el trabajo, use la función expansión: escriba **mv A** y pulse la tecla del tabulador. Si no existe ningún otro archivo que comience con esa inicial, la shell le mostrará la cadena Archivoprueba; en caso contrario, debe introducir más letras y confirmar con la tecla del tabulador. Deje un espacio en blanco después de la primera secuencia (**mv Archivoprueba**), escriba **prueba** y pulse **Return**.

7 Si ahora ejecuta el comando **ls**, ya no se mostrará el Archivoprueba.

8 Para comprobar que se ha producido la modificación, vaya al directorio prueba con el comando **cd prueba**. Escriba **ls**; ahora debe aparecer el Archivoprueba. Con el comando **cd** change directory siempre puede volver a su directorio privado.

9 Si necesita una copia del archivo, utilice el comando **cp** copy. Para copiar Archivoprueba en Archivoseguridad, escriba **cp Archivoprueba Archivoseguridad**. Para ver ambos archivos, ejecute **ls**.

Indicación de la ruta

Para trabajar con archivos o directorios siempre debe indicar la ruta correcta. Para ello no siempre es necesario que indique la ruta completa (absoluta) desde el directorio raíz hasta el archivo deseado. Puede hacerlo desde el directorio en el que se encuentre actualmente. Esto significa que dispone de varias formas de mostrar el archivo Archivoprueba en el directorio prueba: ruta relativa con **ls prueba/*** o ruta absoluta con **ls ~/prueba/***.

Para ver los directorios personales de otros usuarios, escriba **ls ~** seguido del nombre de usuario. En la estructura de directorios mencionada aquí existía un

usuario inventado tux. Con el comando **ls ~Javier** puede ver el contenido del directorio personal del usuario Javier.

El directorio actual se representa con un punto y el directorio inmediatamente superior, con dos. Así, escribiendo **ls ..** se le muestra el contenido del directorio madre del actual directorio; con **ls ../..** sube dos escalones en la jerarquía de directorios.

Ejemplo: El trabajo con rutas

Este otro ejemplo le aclarará cómo moverse por la estructura de directorios de su sistema Linux.

- 1 Para estar seguro de que se encuentra en su directorio personal, introduzca **cd** en la consola. Aquí tiene el subdirectorio prueba. Ahora creará un subdirectorio adicional con el nombre prueba2. Introduzca para ello el comando **mkdir prueba2**.
- 2 Cambie de ubicación a prueba2 con el comando **cd prueba2** y cree allí un nuevo directorio de nombre Subdirectorio. Cambie a éste utilizando la función expansión: es decir, escriba sólo **cd Su** y pulse la tecla **Tab**. El resto de la palabra será añadida automáticamente.
- 3 Ahora se encuentra en Subdirectorio. A continuación moveremos el archivo Archivoseguridad creado anteriormente a este directorio. Para ello, no hace falta que se desplace de nuevo al directorio prueba; bastará con introducir la ruta relativa al archivo deseado. Deberá introducir el siguiente comando: **mv ../../prueba/Archivoseguridad .** (no se olvide del punto final ya que de esa manera define el directorio actual como destino final para el comando **mv**). Con cada dos puntos accederá a un nivel superior en la estructura de directorios, por lo que en este caso ../../, hace referencia a su directorio personal.

Comodines

Otra ventaja de la shell son los llamados comodines o wildcards.

? : Sustituye exactamente un carácter.

***** : Sustituye una cantidad cualquiera de caracteres.

[set] : Reemplaza sólo un carácter de los que forman la secuencia de caracteres

escrita entre corchetes, aquí representado con la cadena de caracteres set.

[!set] : Representa sólo un símbolo cualquiera, exceptuando los incluidos en la

cadena set.

Supongamos que tiene en su directorio prueba los archivos Archivoprueba, Archivoprueba1, Archivoprueba2 y archiPrueba. En este caso, con el comando **ls Archivoprueba?** se le mostrarán los archivos: Archivoprueba1 y Archivoprueba2. Con **ls Archivo*** se le enseñará también el archivo Archivoprueba.

Con **ls *chi*** obtendrá todos los archivos de prueba. Mientras que con el comodín set puede pedir que se muestre sólo los archivos que terminen por un número: **ls Archivoprueba[1-9]**.

El más poderoso siempre es el comodín *****. Con él se puede, por ejemplo, copiar todos los archivos de un directorio en otro, o bien borrarlos. Así, el comando **rm *pru*** borraría de su directorio prueba todos los archivos que contuvieran en su nombre la secuencia de caracteres **pru**.

Less y More

Dos pequeños programas le permiten visualizar archivos de texto directamente en la shell. Por tanto, no necesita abrir primero un editor si quiere leer, por ejemplo, un archivo llamado Leame.txt. Tan solo tiene que introducir el

comando: **less Leame.txt**. De esa forma podrá visualizar el texto en la ventana de la consola. Con la **barra espaciadora** puede avanzar de página en página, y con las teclas de los cursores puede moverse en el texto hacia adelante y hacia atrás. Para terminar **less** utilice la tecla **q**.

El nombre **less** tiene su origen en el dicho a veces menos es más. El programa **less** no sólo se puede usar para ver archivos de texto, sino que también puede servir para ver cómodamente la salida de comandos . Esto se explicará en el siguiente punto.

Además de **less** puede usar también el programa **more**, algo más antiguo y que resulta un poco menos cómodo, ya que con él no puede hojear el texto.

Desvíos

Normalmente la salida estándar de la shell es la pantalla (o bien la ventana de la shell) y la entrada estándar es el teclado. Para desviar la salida de un comando a un programa como **less**, tiene que utilizar una tubería (pipe).

Para ver los archivos dentro del directorio prueba, introduzca el comando **ls prueba | less**. Para obtener el símbolo **|**, pulse simultáneamente **Alt Gr** y la tecla con el signo de exclamación (!) y el número uno (1).

Este procedimiento vale para la distribución de teclado de España y no necesariamente para el teclado latinoamericano. La consola muestra ahora el contenido del directorio prueba utilizando **less**. Esto sólo es razonable, si la lista del contenido del directorio no cabe en la ventana. Por ejemplo con el comando **ls /dev** sólo puede ver algunos de los ficheros en ese directorio, mientras que **ls /dev | less** permite visualizar todo.

Para grabar la salida de un comando en un fichero, el comando equivalente sería: **ls test > contenido**. El resultado es un fichero nuevo con el nombre contenido, que contiene el listado de los ficheros y directorios en prueba. Mediante **less contenido** es posible visualizar el fichero.

Por otra parte es posible utilizar un fichero como entrada a un comando. Por ejemplo puede ordenar alfabéticamente las líneas que haya escrito en el fichero de prueba mediante **sort < Archivoprueba**. La salida de este comando aparece en la pantalla y se ve el texto anteriormente escrito ordenado según las letras iniciales de cada línea. Esto sirve por ejemplo para ordenar una lista de nombres.

Para crear un archivo nuevo con la lista ordenada hay que desviar el resultado del comando **sort** a un archivo nuevo. Para realizarlo, puede crear en un editor una lista no ordenada y grabarla bajo el nombre lista en el directorio prueba. Entre al directorio prueba y ejecute el comando **sort < lista > ListaOrdenada**. Visualice la lista nueva con **less**.

La salida estándar de errores es igualmente la pantalla. Pero si quiere desviarla a un archivo con el nombre Error, escriba **2> Error** a continuación de la orden. Si escribe **>& Salida** después de un comando, tanto la salida estándar como la salida de errores se guardarán en el archivo Salida. Si en lugar de un **>**, emplea dos **>>**, la salida se añadirá a un archivo existente.

Archivar y comprimir datos

Después de haber creado algunos archivos y directorios, veremos ahora el tema de archivar y comprimir datos. Supongamos que desea empaquetar todo el directorio prueba en un archivo, ya sea para guardarlo en un disquete como copia de seguridad o para enviarlo por correo electrónico. Para ello, use el comando **tar** tape archiver. Con **tar --help** puede ver todas las opciones de **tar**. A continuación, se aclararán las más importantes:

- c** : create crear un archivo nuevo.

- t** : table ver el contenido de un archivo.

- x** : extract descomprimir un archivo.

- v** : verbose ver todos los archivos en la pantalla durante el proceso de compresión.

- f** : file con esta opción puede dar un nombre al archivo de empaquetamiento.

Al comprimir, esta opción ha de introducirse en último lugar.

Para empaquetar ahora el directorio prueba con todos sus archivos y subdirectorios en un archivo con el nombre prueba.tar, necesitaremos en cualquier caso las opciones: **-c** y **-f**. Si queremos hacer un seguimiento del proceso de empaquetado usaremos también la opción **-v**. Con **cd** diríjase primero a su directorio personal, donde se encuentra el directorio prueba. Introduzca ahora el comando **tar -cvf prueba.tar prueba**.

Vea después el contenido del archivo empaquetado con: **tar -tf prueba.tar**. A propósito, el directorio prueba con todos sus archivos y directorios no sufre ningún cambio y sigue estando en su disco duro. Para desempaquetar por completo el archivo, introduzca el comando: **tar -xvf prueba.tar**. No haga esto aún ya que antes comprimiremos el archivo para ahorrar espacio.

Para ello, se aconseja utilizar el famoso programa **gzip**. Introduzca: **gzip prueba.tar**. Si ahora ejecuta **ls**, verá que el archivo prueba.tar ha desaparecido y en su lugar existe otro archivo de nombre prueba.tar.gz. Este archivo es mucho más pequeño, y por lo tanto más apropiado para ser enviado por e-mail o para que quepa en un disquete.

Ahora proceda a desempaquetar el archivo en el directorio prueba2 creado anteriormente. Para ello, copie el archivo en el directorio prueba2 con el comando **cp prueba.tar.gz prueba2**. Introduzca seguidamente **cd prueba2**. Para descomprimir un archivo .tar.gz utilice **gunzip**. La orden que se debe ejecutar es: **gunzip prueba.tar.gz**, con lo que obtendrá nuevamente un archivo prueba.tar.

Ahora sólo queda desempaquetar este archivo de la siguiente manera: **tar -xvf prueba.tar**. Con el comando **ls** puede comprobar que ahora tiene un nuevo directorio prueba con el mismo contenido que el directorio prueba que se encuentra en su directorio personal.

mtools

Los **mtools** son una serie de comandos con los que se puede trabajar en un sistema de archivos MS-DOS. Esto es sobre todo importante para trabajar directamente con la disquetera. Como en MS-DOS, a la disquetera se la puede denominar a:

Los comandos reciben los mismos nombres que en MS-DOS, sólo que van precedidos de una m:

mdir a: : le muestra el contenido del disquete introducido en la disquetera a:.

mcopy Archivoprueba a: : copia el archivo Archivoprueba al disquete.

mdel a:Archivoprueba : borra Archivoprueba de a:

mformat a: formatea los disquetes a alto nivel; para ello el disquete tiene que ser

previamente formateado a bajo nivel con el comando

fdformat /dev/fd0h1440 o estar ya formateado.

mcd a: : a: será su directorio actual.

mmd a:prueba : crea el subdirectorio prueba en el disquete.

mrd a:prueba : borra el subdirectorio prueba del disquete.

Hacer limpieza

Para que nuestro directorio personal quede un poco más ordenado, puede borrar todos los directorios y archivos de prueba con los comandos **rm** y **rmdir**.

7.3.- Derechos de usuario

Linux fue concebido desde un principio como un sistema multiusuario, es decir, para ser usado por varios usuarios al mismo tiempo. Esta es una de las grandes diferencias con el sistema operativo Windows de Microsoft.

La característica más importante es la obligación de registrarse en el sistema al inicio de cada sesión de trabajo. Para ello el usuario dispone de un nombre de

usuario propio con su contraseña correspondiente. Esta diferenciación entre usuarios impide que personas no autorizadas puedan acceder a archivos que les han sido vetados. Un usuario, por lo general, no puede (o sólo de forma muy limitada) hacer grandes cambios en el sistema como, por ejemplo, la instalación de programas.

Tan sólo el superusuario, o administrador, root dispone prácticamente de derechos ilimitados y posee acceso sin restricciones a todos los archivos. Quien utilice este concepto con cautela y sólo se registre como el todopoderoso root cuando sea realmente necesario, puede reducir en gran medida el riesgo de perder datos de forma involuntaria.

La amenaza de caballos de Troya o de comandos destructivos introducidos sin querer se reducen de forma significativa dado que normalmente sólo el superusuario puede borrar archivos del sistema o formatear discos duros.

Derechos en el sistema de archivos

Fundamentalmente, en el sistema de archivos Linux, cada archivo pertenece a un usuario y a un grupo. Cada uno de estos dos tipos de propietarios, así como también los extraños, pueden obtener derechos de escritura, de lectura así como de ejecución por parte del propietario del archivo.

Se denomina grupo a un conjunto de usuarios con un tipo concreto de derechos colectivos: Un ejemplo podría ser un grupo que trabajase en un proyecto concreto al que podríamos llamar proyecto3. En Linux, todos los usuarios son miembros de al menos una de estas unidades de administración. El grupo por defecto es users. Pueden crearse casi tantos grupos adicionales como se desee, si bien, esta función está reservada para root. Todos los usuarios pueden averiguar de qué grupo son miembros con ayuda del comando **groups**.

Derechos sobre archivos

A continuación observaremos más de cerca la estructura de los derechos en el sistema de archivos. El resultado de ejecutar **ls -l** podría ser lo que se ve en la salida de pantalla del ejemplo siguiente:

Ejemplo: Muestra de los derechos de acceso a un archivo

`-rw-r----- 1 Javier proyecto3 14197 Jun 21 15:03 Roadmap`

Tal y como se puede observar en la tercera y cuarta columna, este archivo pertenece al usuario Javier, y al grupo proyecto3. Para poder ver detalladamente los derechos sobre el archivo, hay que mirar con atención la primera columna.

| - | rw- | r-- | --- |
|------|---------------------|-------------------|----------------------------|
| Tipo | Derechos de usuario | Derechos de grupo | Derechos de otros usuarios |

Esta columna consta de un carácter inicial y de otros nueve caracteres divididos en grupos de tres. La primera de las diez letras representa el tipo de archivo. El guión - nos indica que en este caso se trata de un archivo. Podría tratarse también de un directorio (d), un enlace (l), un dispositivo de bloques (b) o un dispositivo de caracteres (c).

Los tres bloques siguientes siguen un esquema común: El primer carácter de cada uno de los tres grupos indica si los diferentes grupos de usuarios tienen derecho de lectura sobre este archivo (r) o si no lo tienen (-). Los dos signos siguientes funcionan de manera similar: el carácter (w) situado en el medio simboliza el derecho a escribir en el objeto en cuestión; esto no sería posible si aparece un guión (-).

Por otra parte, en la columna de la derecha de cada grupo puede aparecer el carácter (x) que indica el permiso de ejecución. Dado que en este caso se trata de un archivo de texto y no de un programa ejecutable, se puede prescindir perfectamente de este permiso de ejecución.

En este ejemplo Javier, como dueño del archivo Roadmap, tiene tanto el permiso de lectura (r) como el de escritura (w), pero no puede ejecutarlo (no hay x). Los miembros del grupo proyecto3 sólo tienen permiso de lectura sobre el archivo, pero no de escritura ni de ejecución. El resto de usuarios no tienen

ningún tipo de permiso sobre este archivo. Se puede seguir concediendo más permisos mediante las llamadas ACLs Access Control Lists.

Permisos sobre directorios

Ahora nos dedicaremos a los permisos de acceso sobre los directorios, representados por el carácter d. Los permisos tienen aquí un significado algo diferente. Sirva de aclaración el siguiente ejemplo:

Ejemplo: Ejemplo de derechos de acceso a un directorio

```
drwxrwxr-x 1 Javier proyecto3 35 Jun 21 15:15 Datosproyecto
```

En la salida de pantalla es fácil reconocer que el propietario de (Datosproyecto) es (Javier) y que el grupo de propietarios es (proyecto3). A diferencia de los permisos de archivos de Derechos sobre archivos, el derecho de lectura (r) indica aquí que se puede ver el contenido del directorio.

El permiso de escritura (w) hace referencia a la posibilidad de añadir nuevos archivos en este directorio y el derecho de ejecución (x) autoriza a cambiar a este directorio. En relación al ejemplo anterior, esto significa que tanto el usuario Javier como los miembros del grupo proyecto3 tienen derecho a cambiar al directorio Datosproyecto (x), ver su contenido (r), y añadir archivos (w).

Los demás usuarios tienen menos derechos: sólo pueden acceder a él (x) y echar un vistazo (r), pero no pueden añadir ningún archivo (falta w).

Modificar los permisos sobre archivos

Cambiar derechos de acceso

Los permisos de acceso de un archivo o de un directorio pueden ser modificados por el propietario (y por supuesto también por root) mediante el comando **chmod**, que se introduce junto con los parámetros de permisos de acceso que se desean cambiar y con el nombre del archivo a modificar.

Los dos parámetros están formados por:

1. las categorías afectadas

- **u user** - el propietario del archivo
- **g group** - el grupo al que pertenece el propietario del archivo
- **others** - otros usuarios (si no se introduce ningún parámetro, los cambios

afectarán a todas las categorías)

2. un carácter para retirar permisos (-), dejar igual (=) o añadir (+)

3. las abreviaturas ya conocidas para

- **r** read - lectura
- **w** write - escritura
- **x** execute – ejecución

4. así como el nombre del archivo (o archivos) correspondiente escrito a continuación y separado por un espacio en blanco.

Si por ejemplo, el usuario Javier, del ejemplo anterior, desea otorgar el permiso de escritura (w) sobre el directorio Datosproyecto a otros usuarios, lo puede llevar a cabo usando el comando: **chmod o+w Datosproyecto**.

Sin embargo, para quitar el derecho de escritura a todos los usuarios aparte de sí mismo, utilice el comando **chmod go-w Datosproyecto**. Para prohibir a todos los usuarios crear un archivo en el directorio Datosproyecto introduzca **chmod -w Datosproyecto**. Ahora ni siquiera el propietario del archivo tiene derecho de modificar su archivo antes de que restablezca el derecho de escritura.

Cambiar los derechos de propiedad

Otros comandos importantes al respecto son **chown** (Change Owner) y **chgrp** (Change Group), que gestionan las relaciones de propiedad de los elementos del sistema de archivos. El comando **chown** permite cambiar el propietario de

un archivo. No obstante, esta modificación sólo puede ser realizada por root.

Supongamos que el archivo Roadmap del ejemplo anterior ya no debe pertenecer a tux, sino al usuario geeko. El comando que root debe utilizar para esta modificación es: **chown geeko Roadmap**.

De forma análoga, el comando **chgrp** permite cambiar el grupo de propietarios. Hay que tener en cuenta que el propietario del archivo debe pertenecer al grupo al que desea otorgar la categoría de grupo de propietarios. Así por ejemplo, mediante el siguiente comando, el usuario Javier del ejemplo puede cambiar el grupo de propietarios del archivo Datosproyecto a proyecto4 utilizando el comando **chgrp proyecto4 Datosproyecto**, siempre y cuando él sea miembro de este grupo. Para root no existe esta restricción.

El Setuid-Bit

Hay situaciones en las que los derechos de acceso son demasiado restrictivos. En estos casos, existen en Linux opciones adicionales que permiten cambiar temporalmente el usuario y grupo de identidad actual para realizar una determinada acción.

Por ejemplo, el programa **cdrecord**, empleado para escribir CDs (o DVDs) requiere derechos de root para acceder al grabador. Por tanto, un usuario normal no puede grabar CDs. Sin embargo, sería peligroso que todos los usuarios tuvieran acceso directo a todos los dispositivos.

La solución se encuentra en el mecanismo Setuid. Setuid (Set User ID) es un atributo especial de archivo que el sistema regula, con el fin de que se ejecuten los programas seleccionados con una determinada identificación de usuario previamente indicada. Veamos el comando **cdrecord**:

```
-rwxr-x--- 1 root root 281356 2002-10-08 21:30 /usr/bin/cdrecord
```

Asignamos el Setuid Bit con el comando **chmod u+s /usr/bin/cdrecord**.

Mediante **chgrp users /usr/bin/cdrecord** el programa **cdrecord** se asigna al

grupo users. Los derechos de acceso que tenemos:

```
-rws--x--- 1 root users 281356 2002-10-08 21:30 /usr/bin/cdrecord
```

Con Setuid-Bit, todos los usuarios que pertenezcan al grupo users pueden utilizar el programa. En la práctica ésto significa que el programa se puede ejecutar como usuario root.

Hay que tener en cuenta que cada vez que otorga a un programa el Setuid-Bit, existe una nueva posibilidad de acceder malintencionadamente al computador, por ello debemos realizar este proceso sólo en casos excepcionales, cuando conozca el programa con toda exactitud y pueda calcular los riesgos que corre.

El Setgid-Bit

El atributo Setuid es válido para el usuario, pero el atributo Setgid tiene las mismas propiedades para el grupo. En este caso, se ejecutará un programa con la identificación de grupo que se indicó, independientemente del usuario que lo arrancó. Por eso todos los archivos y subdirectorios que se crean dentro de este directorio pertenecen al mismo grupo que el directorio:

```
drwxrwxr-- 2 root archive 48 Nov 19 17:12 backup
```

Ahora asignamos el Setgid-Bit mediante el comando **chmod g+s /prueba**

A continuación, los derechos de acceso serán:

```
drwxrwsr-- 2 root archive 48 Nov 19 17:12 backup
```

El Sticky-Bit

Además del Setuid-Bit y el Setgid-Bit, existe el denominado Sticky-Bit. Aquí se debe diferenciar si se trata de un programa ejecutable o de un directorio. Hoy en día, este Bit no se utiliza para archivos por lo que solo tiene un significado histórico.

Pero si se asigna este atributo a un directorio, éste impide que el usuario borre sus archivos. (En directorios con Sticky-Bit, los usuarios sólo puede eliminar archivos que posean ellos mismos). Los ejemplos típicos son los directorios /tmp y /var/tmp:

```
drwxrwxrwt 2 root root 1160 2002-11-19 17:15 /tmp
```

Access Control Lists

Una extensión del concepto tradicional de derechos sobre archivos y directorios es el concepto de las listas de control de acceso ACL Access Control Lists. Mediante estas listas es posible otorgar derechos de acceso a usuarios y grupos, distintos de los usuarios y grupos que son dueños de los objetos.

Los directorios y archivos con derechos extendidos se reconocen con un sencillo **ls -l** por lo siguiente:

```
-rw-r--r--+ 1 tux proyecto3 517 2003-01-08 18:12 Roadmap
```

A primera vista el resultado del comando **ls** a penas ha cambiado. El dueño del archivo Roadmap es el usuario Javier que forma parte del grupo proyecto3. Javier tiene tanto derechos de lectura como de escritura sobre este archivo; el grupo y el resto del mundo tienen derecho de lectura.

El único indicador de un cambio respecto a un archivo sin ACL es el símbolo + en la primera columna, junto a los bits de derechos.

Para obtener detalles sobre la ACL del fichero de ejemplo Roadmap, utilice el comando **getfacl**:

```
# file: Roadmap
# owner: Javier
```

```
# group: proyecto3
user::rw-
user:jose:rw-      effective: r--
group::r--
group:djungla:rw-  effective: r--
mask::r--
other::---
```

Las primeras tres líneas no dan ninguna información que no se pueda obtener con **ls -l**, ya que sólo se trata del nombre de archivo, del dueño y del grupo.

Las líneas comprendidas entre la 4 y la 9 representan los verdaderos parámetros de la ACL, ACL entries. Los derechos habituales son un subconjunto de los derechos que se pueden definir con la ACL.

El ejemplo de ACL otorga derechos de lectura y escritura al dueño del archivo y al usuario jose (líneas 4 y 5). Lo mismo vale para los grupos: El grupo del dueño del archivo tiene derecho de lectura (línea 6) y el grupo djungla tiene derechos de lectura y escritura.

El parámetro mask en línea 8 limita los derechos del usuario jose y del grupo djungla a solo lectura. Todos los demás usuarios y grupos no tienen derecho de lectura (línea 9).

7.4.- Resumen de los comandos más importantes de Linux

En este apartado se ofrece un resumen de los comandos más importantes de Linux. Además del significado básico de cada comando, dando una explicación de algunos de los parámetros y, si es pertinente, un ejemplo típico de uso.

Para obtener más información sobre los diferentes comandos, utilice el comando **man** seguido del nombre del comando, por ejemplo: **man ls**.

Ejecutando el comando **man man** puede obtener más información sobre el

propio comando **man**.

Tendrá que adaptar las especificaciones de los comandos a sus propias necesidades. No tiene sentido escribir **ls archivo(s)** si no existe en realidad ningún archivo de nombre archivo(s). Por lo general puede combinar más de un parámetro, escribiendo por ejemplo **ls -la** en lugar de **ls -l -a**.

Comandos del sistema

Administración de archivos

ls [opcion(es)] [archivo(s)]

Si introduce **ls** sin más datos ni parámetros, podrá ver un listado resumido con el contenido del directorio en el que se encuentre en ese momento.

-l : Lista detallada.

-a : Para ver los archivos ocultos.

cp [opción(es)] archivo_original archivo_copia

Crea una copia de archivo original en archivo copia.

-i : Espera, dado el caso, una confirmación antes de sobrescribir un archivo copia existente.

-r : Copia recursiva (incluyendo subdirectorios).

mv [opción(es)] archivo_origen archivo_destino

Hace una copia de archivo origen en archivo destino y a continuación borra el archivo original.

-b : Crea una copia de seguridad antes de desplazar archivo_origen.

-i : Espera, dado el caso, una confirmación antes de sobrescribir un archivo destino existente.

rm [opcion(es)] archivo(s)

Borra el archivo(s) indicado del sistema de archivos. Los directorios no serán

borrados con el comando **rm** a menos que se haya especificado de forma explícita a través del parámetro **-r**.

-r : Borra, si existen, los subdirectorios existentes.

-i : Espera una confirmación antes de borrar cada uno de los archivos.

ln [opción(es)] archivo_origen archivo_destino

Crea en archivo destino una referencia cruzada dentro del sistema de archivos, (Link) al archivo origen con otro nombre. Normalmente este enlace apunta directamente a la posición de archivo origen dentro de un sistema de archivos.

Si por el contrario se ejecuta el comando **ln** con la opción **-s**, se creará un enlace simbólico que apuntará exclusivamente a la rama de archivo origen y que por esa razón se podrá usar fuera de los límites del sistema de archivos.

-s : Crea un enlace simbólico.

cd [opción(es)] directorio

Cambia el directorio actual. Si sólo se introduce **cd**, se desplazará al directorio home.

mkdir opción(es) directorio

Crea un directorio nuevo.

rmdir [opcion(es)] directorio

Borra el directorio, pero sólo si está vacío.

chown [opcion(es)] nombreusuario.grupo archivo(s)

Cambia el dueño de un archivo, otorgándole la propiedad a nombreusuario.

-R : Cambia los archivos y directorios en todos los subdirectorios.

chgrp [opcion(es)] nombregroupo archivo(s)

Cambia el nombre del grupo al que pertenece el archivo por nombregroupo. El dueño del archivo solo puede cambiar este valor si pertenece tanto al grupo al que pertenecía el archivo hasta ahora como al nuevo grupo al que pertenecerá en adelante.

chmod [opciones] modo archivo(s)

Cambia los permisos de acceso a uno o varios archivos. El parámetro modo tiene varias partes: grupo y tipo de acceso. Grupo acepta los siguientes caracteres:

u : Para el usuario *user*,

g : Para el grupo *group*,

o : Para todos los demás *others*.

+ y **-** se utilizan para permitir o denegar el acceso, de manera que:
el símbolo **+** permitirá el acceso y el símbolo **-** lo denegará.

Los tipos de acceso son controlados por las siguientes opciones:

r : Para lectura *read*,

w : Para escritura *write*,

x : Ejecutar archivos o cambiar de directorio *execute*.

s : Setuid-Bit; el programa se ejecutará con los privilegios del propietario del archivo.

gzip [parámetros] archivo(s)

Este programa comprime el contenido de archivos mediante unos complicados procesos matemáticos. El nombre del archivo comprimido termina con **.gz** y se debe descomprimir antes de ser usado nuevamente. Si desea comprimir más de un archivo o directorios completos, deberá usar además el comando **tar**.

-d : Descomprime el archivo comprimido **gzip**, recuperando éste su tamaño original y pudiendo volver a ser utilizado de forma normal (corresponde al comando **gunzip**).

tar opciones nombrepaquete archivo(s)

El comando **tar** agrupa uno o (normalmente) más archivos en un llamado paquete, que luego se puede p.ej. comprimir.

tar es un comando muy complejo, que ofrece una gran cantidad de opciones. A continuación se presentan algunas de las combinaciones más usadas.

-f : La información generada por el proceso se escribe en un archivo y no

se muestra en la pantalla, como está previsto por defecto file:

- c** : Crea un nuevo archivo tar *create*.
- r** : Agrega archivos a un paquete existente.
- t** : Muestra el contenido de un paquete.
- u** : Agrega archivos al paquete, pero sólo si estos son más recientes que los existentes.
- x** : Desempaqueta archivos de un paquete (extrae)
- z** : Comprime con gzip el paquete generado.
- j** : Comprime con bzip2 el paquete generado.
- v** : Muestra los nombres de los archivos procesados.

Los archivos creados mediante **tar** terminan en .tar. Cuando el archivo tar ha sido además comprimido con **gzip**, la terminación será .tar.gz, con bzip2, será .tar.bz2.

locate prueba

Con el comando **locate** se puede averiguar en qué directorio se encuentra el archivo especificado. Para ello también se puede usar los comodines. El programa trabaja muy rápido ya que no busca en el sistema de archivos, sino en una base de datos propia creada a tal efecto.

Éste es el mayor problema de estos comandos rápidos, ya que los archivos creados posteriormente a la última actualización de esta base de datos, no se tienen en cuenta.

Como root puede crear esta base de datos usando el comando **updatedb**.

updatedb [opción(es)]

Con este comando se puede actualizar de forma sencilla la base de datos requerida por **locate**. Para conseguir que todos los archivos se tengan en cuenta, se debe ejecutar el comando como root.

También es posible, añadiendo al final del comando el símbolo &, que el programa se ejecute en segundo plano, y así poder seguir trabajando mientras se produce la actualización . (updatedb &).

find [opcion(es)]

Con el comando **find** puede buscar un archivo en un determinado directorio. El primer argumento representa el directorio desde el que debe comenzar la búsqueda. La opción **-name** requiere una cadena de búsqueda, en la que también se permite el uso de comodines. Al contrario que **locate**, **find** busca en el directorio especificado, no en una base de datos.

Comandos para visualizar contenidos

cat [opcion(es)] archivo(s)

Con el comando **cat** se ve el contenido del archivo elegido sin interrupciones.

-n : Numera el texto en el borde izquierdo.

less [opcion(es)] archivo(s)

Con este comando se puede examinar el contenido del archivo elegido. Por ejemplo puede subir y bajar media hoja de pantalla con las teclas “flecha arriba” y “flecha abajo” con la **barra espaciadora** hasta una página entera hacia adelante. También puede desplazarse al comienzo y al final del texto con las teclas **Inicio** y **Fin**. Y con la tecla **Q** puede cerrar este modo de visualización.

grep [opción(es)] términodebúsqueda archivo(s)

grep está pensado para buscar un determinado término de búsqueda en el archivo(s) elegido. Si la búsqueda se ha realizado con éxito, el programa mostrará la línea en la que se ha encontrado el término de búsqueda, así como el nombre del archivo.

-i : Ignora las diferencias entre mayúsculas y minúsculas.

-l : Da sólo el nombre del archivo correspondiente y no el número de línea.

-n : Numera, además, las líneas donde se encontró el texto buscado.

-L : Da una lista de los archivos que no contengan el término de búsqueda.

diff [opcion(es)] archivo1 archivo2

diff fue creado para comparar el contenido de dos archivos cualesquiera y mostrar en una lista las líneas en las que se hayan detectado diferencias.

Los programadores utilizan con frecuencia este comando para enviar los cambios realizados en sus programas sin necesidad de enviar todo el texto fuente.

-q : Indica tan sólo si existen diferencias entre el contenido de los dos archivos seleccionados.

Sistema de archivos

mount [opcion(es)] [dispositivo] punto_de_montaje

Con ayuda de este comando se puede montar diferentes dispositivos de almacenamiento. Con ello se entiende la unión de discos duros, CD-ROM y otras unidades de disco con un directorio del sistema de archivos Linux.

-r : monta el dispositivo como sólo lectura *read only*.

-t : sistema de archivos : Especifica el sistema de archivos.

Los más comunes son: **ext2** para discos duros de Linux,

msdos para dispositivos MS-DOS, **vfat** o **ntfs** para

sistema de archivos de particiones Windows e **iso9660** para CDs.

En el caso de unidades que no estén definidas en el archivo `/etc/fstab`, se deberá también especificar el tipo de dispositivo. En este caso, sólo root podrá montarlo. Para que el sistema de archivos pueda ser montado por otros usuarios, introduzca la opción **user** en la línea correspondiente del archivo `/etc/fstab` (separado por comas) y guarde los cambios.

umount [opcion(es)] puntodemontaje

Con este comando se desmonta una unidad de disco del sistema de archivos.

Ejecute este comando antes de retirar un medio de almacenamiento de la unidad de discos. ¡De lo contrario se expone a una pérdida de datos! Por norma general, tanto el comando **mount** como el comando **umount** sólo pueden ser ejecutados por root. Excepción: Cuando, para el dispositivo, aparece la opción **user** en el archivo `/etc/fstab`.

Información

df [opcion(es)] [directorio]

Si se ejecuta el comando **df** *disk free* sin ninguna opción, se nos dará una estadística sobre el espacio total, el ocupado y el libre de todas las unidades de disco montadas. Si por el contrario se introduce un directorio, se mostrarán sólo las estadísticas de la unidad de disco que incluye dicho directorio.

-H : Nos enseña el número de bloques ocupados en gigabytes, megabytes o kilobytes ? de forma legible para el ser humano *human readable*.

-t : Tipo de unidad (ext2, nfs, etc.).

du [opcion(es)] [ruta]

Este comando, usado sin opciones, indica el espacio total de almacenamiento utilizado por todos los archivos del directorio en el que nos encontremos. Si existen subdirectorios en él, también se indicará su tamaño total.

-a : Indica el tamaño de cada uno de los archivos.

-h : Listado en forma legible para el ser humano.

-s : Nos enseña tan sólo el tamaño total.

free [opcion(es)]

free nos muestra el tamaño total y el uso de memoria física y de intercambio.

-b : Indicado en bytes.

-k : Indicado en kilobytes.

-m : Indicado en megabytes.

date [opcion(es)]

Ejecutando este pequeño programa, se nos informará de la hora del sistema. Además como root, se puede modificar con este comando la hora del sistema.

Procesos

top [opcion(es)]

Por medio de top obtendrá una lista rápida de todos los Procesos que se estén ejecutando. Pulsando la tecla **h** se le enseñara una página con aclaraciones y con las opciones más importantes para adecuar este programa a sus necesidades.

ps [opcion(es)] [proceso-ID]

Si se ejecuta sin opciones se mostrarán en una tabla todos los procesos propios, es decir, todos los programas y procesos que haya inicializado uno mismo. Tenga cuidado al usar las opciones de este comando; pues no se debe anteponer un guión.

aux : Proporciona una lista detallada de todos los procesos, sin tener en cuenta el dueño.

kill [opcion(es)] proceso-ID

A veces, desafortunadamente, nos encontramos con programas que no se pueden cerrar de forma normal. Con el comando **kill**, se puede eliminar todos los procesos muertos con ayuda del ID del proceso.

Para ello envía una señal llamada TERM que ordena al programa cerrarse a sí mismo. Si esto tampoco sirve tenemos aún otro parámetro de gran ayuda:

-9 : Envía en lugar de una señal tipo TERM una señal tipo KILL, lo que ocasiona que el proceso sea liquidado por el sistema operativo, consiguiendo en casi todos los casos que el proceso especificado se termine.

killall [opcion(es)] nombreproceso

Este comando funciona de forma equivalente a **kill**, sólo que para este comando basta con dar el nombre del proceso - y no su ID - para matar todos los procesos que tengan ese nombre.

Red

ping [opcion(es)] nombre_computadora/direcciónIP

ping es el comando por excelencia para comprobar que las funciones básicas de una red TCP/IP funcionan correctamente. La herramienta manda a otro ordenador un pequeño paquete de datos, ordenándole que una vez recibido lo devuelva de inmediato.

Si esto funciona, **ping** se lo indicará con un mensaje, con lo que se asegura la capacidad de transmisión básica de la red.

-c número : Determina el número total de paquetes enviados, tras lo cual el programa se cierra. No hay limitaciones por defecto.

-f : Envía tantos paquetes de datos como sea posible. Comando usado para probar al límite la capacidad de una red, pero que sin embargo sólo puede ser usado por root.

-i valor : Segundos transcurridos entre el envío de dos paquetes de datos; el valor predeterminado es un segundo.

nslookup

Para transformar nombres de dominios en direcciones IP. Con esta herramienta puede dirigir preguntas al servicio de información correspondiente (servidor DNS).

telnet [opcion(es)] nombre_computadora o direcciónIP

Telnet es un protocolo de Internet que permite trabajar en otros ordenadores vía red de forma remota. Además Telnet es el nombre de un programa Linux que utiliza este mismo protocolo para permitir el trabajo con otras máquinas sin tener que estar sentado delante de ellas.

Procure no utilizar Telnet en redes en las que pueda ser espiado por terceros. En Internet se deberían usar sobre todo métodos de transmisión cifrados como p.ej. **ssh** y así impedir el peligro que conlleva un uso malintencionado de su contraseña.

Otros

passwd [opcion(es)] [nombre_de_usuario]

Con este comando, cada usuario tiene en todo momento la posibilidad de cambiar su propia contraseña. El superusuario root tiene además permiso para cambiar la contraseña de cualquier usuario.

su [opcion(es)] [nombredeusuario]

su permite cambiar el login del usuario durante una sesión. Si no se utiliza ningún parámetro, la línea de comando exigirá la contraseña de root. Una vez introducida ésta correctamente se obtendrán todos los derechos del superusuario.

También se puede usar el entorno de otro usuario introduciendo detrás del comando el nombre del usuario y posteriormente la contraseña correcta del mismo. root no necesita introducir esta contraseña. Es decir, con los derechos de superusuario se puede asumir sin problemas la identidad de cualquier usuario.

halt [opcion(es)]

Para evitar la pérdida de datos siempre debería apagar su computadora con este programa.

reboot [opcion(es)]

Funciona como el comando **halt**, pero la computadora se volverá a iniciar tras apagarse.

clear

De vez en cuando ocurre que la consola está tan llena de líneas de texto que una limpieza no le iría nada mal. Este comando se usa justamente para eso y no tiene ninguna opción.

Ejercicios

1. Cree un directorio graficas en su espacio personal.

Solución

mkdir graficas

2. Busque una gráfica en el directorio /usr/share/pixmaps y cópiela en el directorio graficas que creó --si lo desea puede verla bien con doble clic desde el administrador de archivos o con el comando **gimp graf** reemplazando graf por el nombre de la gráfica.

Solución

```
cp /usr/share/pixmaps/una.jpg graficas
```

3. Borre la gráfica y el directorio que había creado.

Solución

Sobre el directorio graficas presionar **del**

4. Intente borrar una gráfica del directorio /usr/share/pixmaps (no podrá a menos que usted sea el administrador del sistema, aún así, compruébelo).

Solución

Sobre el directorio /usr/share/pixmaps presionar **del**

5. Con el comando **chroot** ejecute el programa **cat** en un directorio diferente.

Ayuda: Puede revisar las librerías que **cat** emplea con **ldd /bin/cat**

Solución

```
mkdir /root/nj;  
mkdir /root/nj/lib;  
cp /bin/cat /root/nj;  
cp /lib/libc.so.6 /root/nj/lib;  
cp /lib/ld-linux.so.2 /root/nj/lib
```

6. Examine las especificaciones de la CPU que tiene su sistema (Ayuda: comando **more** y un archivo del directorio **/proc**).

Solución

```
cat /proc/cpu/info
```