

Capítulo 7

Permisos de acceso a los archivos

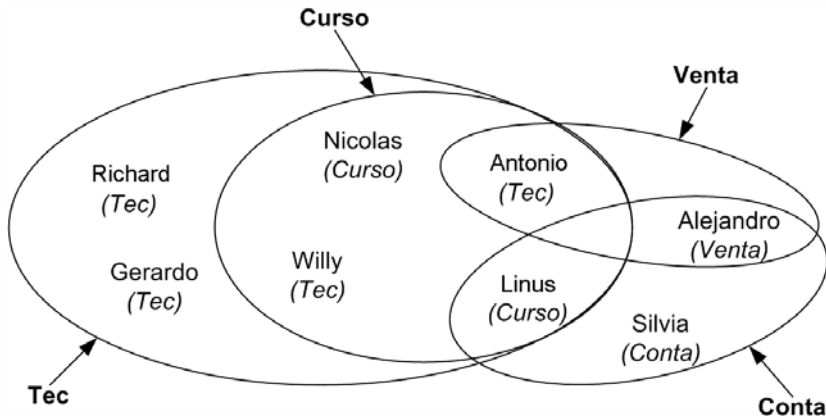
1. Conceptos de cuentas de usuario y de grupos

El sistema GNU/Linux es multiusuario, por tanto las personas que lo usan deben identificarse para asegurar la confidencialidad de los datos contenidos en los archivos. En efecto, no sería aceptable que el usuario "Nicolas" pudiera consultar los archivos personales de "Richard" sin el permiso de este.

Cada una de estas personas dispone por tanto de una "cuenta de usuario" en el sistema; pueden utilizarlo tras ser claramente identificadas. Sin embargo, está permitido compartir archivos entre colaboradores y existe una noción de "grupo de usuarios" en GNU/Linux.

Un usuario debe ser miembro obligatoriamente de un grupo de usuarios en un sistema Unix como GNU/Linux: su grupo principal es el utilizado al crear archivos. Por el contrario, puede pertenecer a otros varios grupos: sus grupos secundarios determinan sus derechos de acceso a los archivos creados por otros miembros de los grupos.

Por ejemplo, si se representa los diferentes servicios de una empresa con su personal, puede observarse que, si bien cada individuo tiene una función primaria (indicada entre paréntesis), algunos pueden asumir varias misiones:



Se observa aquí que:

- Richard y Gerardo pertenecen al servicio técnico (Tec).
- Nicolas, que es ante todo formador (Curso), también forma parte del servicio técnico (Tec).
- Willy, que pertenece al servicio técnico (Tec) principalmente, también trabaja en el departamento de formación (Curso).
- Linus es un formador (Curso) que colabora con los departamentos técnicos (Tec) y de contabilidad (Conta).
- Antonio, del servicio técnico (Tec), ofrece sus competencias al servicio comercial (Venta) y también da clases (Curso).
- Alejandro es un comercial (Venta) que se ocupa también de tareas administrativas (Conta).
- Silvia forma parte únicamente del departamento de contabilidad (Conta).

Para identificar a todos estos usuarios a nivel del sistema operativo, se les atribuye un número único: el UID (*User's ID*); el propietario de un archivo se determina por este número en Unix. Estos usuarios están dotados también de un nombre de usuario único (*login*) y de una contraseña (*password*) para que puedan autenticarse al conectarse al sistema.

De la misma manera, los grupos de usuarios se representan por un nombre único al que se asocia un identificador único: el GID (*Group ID*). Este número se utiliza también para determinar el grupo propietario de un archivo.

1.1 Jerarquía de usuarios

Los usuarios, y por consiguiente las cuentas de usuario, no son todas iguales en Unix. Se pueden distinguir tres tipos de cuentas:

root

Es el usuario más importante del sistema desde el punto de vista de la administración. No se ve afectado por los derechos de acceso a los archivos y puede hacer más o menos de todo en el sistema, excepto escribir en un sistema de archivos montado en lectura exclusiva (CD-ROM). Su UID igual a 0 le confiere su especificidad. Este "superusuario" se encarga de las tareas administrativas del sistema. Para evitar errores al trabajar, es muy recomendable utilizar la cuenta administrativa sólo para las tareas que requieren los derechos de superusuario.

bin, daemon, sync, apache...

Existe en el sistema una serie de cuentas que no se asignan a personas físicas. Estas cuentas sirven para facilitar la administración de los derechos de acceso de ciertas aplicaciones y demonios. Los UID comprendidos entre 1 y 999 se utilizan generalmente para estas cuentas.

linus, nicolas...

Todas las demás cuentas de usuario se asocian a personas reales; su función es permitir a los usuarios estándar conectarse y utilizar los recursos del equipo. El UID de un usuario es normalmente un número superior o igual a 1000.

Observación

Se denominan "demonios" los programas que se ejecutan como tarea en segundo plano, como un servidor web o un servidor de impresión.

Al igual que las cuentas de usuario, existen diferentes tipos de grupos en un sistema GNU/Linux que permiten dar derechos comunes a una serie de usuarios:

root

Su GID es 0 y es el grupo principal del administrador.

bin, daemon, sync, apache...

Estos grupos tienen la misma función que las cuentas del mismo nombre y permiten dar los mismos derechos de acceso a una serie de aplicaciones. Por convención, los grupos del sistema tienen un GID comprendido entre 1 y 999.

curso, tec...

Estos grupos representan una serie de personas reales que deben acceder a los mismos archivos. Típicamente, tienen un GID superior o igual a 1000.

1.2 Comandos útiles

Los comandos **id** y **groups** permiten mostrar información sobre los grupos. El primero da el UID del usuario, el GID de su grupo principal y los GID de todos los grupos a los que pertenece. El segundo sólo proporciona la lista completa de los grupos pero acepta varios nombres de usuario como argumentos:

```
[nicolas]$ whoami
nicolas
[nicolas]$ id
uid=1000(nicolas) gid=1000(curso) grupos=1000(curso),1001(tec)
[nicolas]$ id richard
uid=1002(richard) gid=1001(tec) grupos=1001(tec)
[nicolas]$ groups
curso tec
[nicolas]$ groups gerardo alejandro willy root
antonio : tec curso venta
alejandro : venta conta
willy : tec curso
root : root
```

2. Permisos de Unix

Los permisos de acceso a los archivos determinan las acciones que pueden emprender los usuarios.

■ Observación

La mayoría de los problemas de instalación, configuración y funcionamiento de las aplicaciones en GNU/Linux se debe a derechos de acceso mal adjudicados.

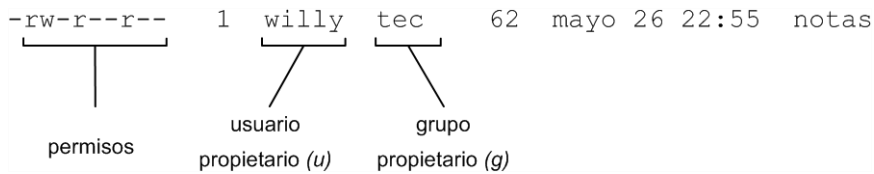
En primer lugar, es necesario saber que los derechos de acceso en Linux se definen por:

- Una cuenta de usuario: propietario del archivo, es en principio el usuario que lo ha creado.
- Un grupo: este grupo es generalmente el grupo principal del propietario del archivo, pero puede ser modificado por este y tomar el valor de uno de sus grupos secundarios.
- Los otros: esta entidad representa toda persona distinta del propietario y que no es miembro del grupo citado anteriormente.

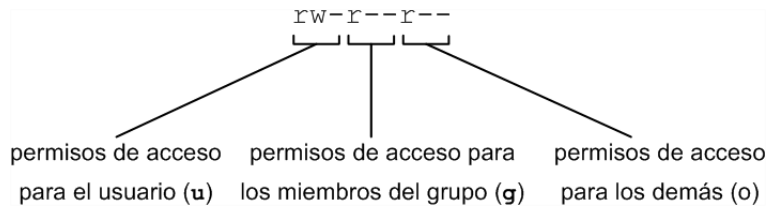
■ Observación

Los derechos de acceso a un archivo se llaman también "modos" en Unix.

Los derechos, el usuario y el grupo propietarios de un archivo pueden verse con el comando **ls -l**:



En este último ejemplo, el archivo pertenece al usuario **willy** y al grupo **tec**; los nueve caracteres **rw-r--r--** definen los derechos de acceso a este archivo para el usuario **willy** (user o **u**), los miembros del grupo **tec** (group o **g**) y los demás (other u **o**). Más exactamente, estos caracteres se distribuyen así:



Todo usuario está asociado, pues, a una de estas entidades para determinar los permisos vigentes.

■ Observación

Atención: si el usuario es propietario del archivo, se aplican los permisos del propietario, y no los del grupo, aunque el usuario sea también miembro de ese grupo.

El comando GNU **ls** puede añadir un carácter adicional a la sucesión de nueve derechos Unix estándar cuando las autorizaciones especiales están ubicadas. Un punto **'** señala un contexto de seguridad SELinux específico y un **+** indica que se utiliza otro método de autorización, como las ACL (*Access Control Lists*).

2.1 Permisos estándar

Los permisos de acceso fundamentales en los archivos y directorios en Unix/Linux son los permisos de lectura **r** (*Read*), escritura **w** (*Write*) y ejecución **x** (*eXecute*).

Estos permisos –definidos para las entidades **u**, **g** y **o**– aparecen en el orden **r**, seguido de **w**, seguido de **x** con el comando **ls -l**. Cuando uno de estos caracteres se reemplaza por un guión, significa que el permiso asociado no está otorgado.

En el ejemplo del apartado anterior, el usuario **willy** que tiene los permisos **rw-**:

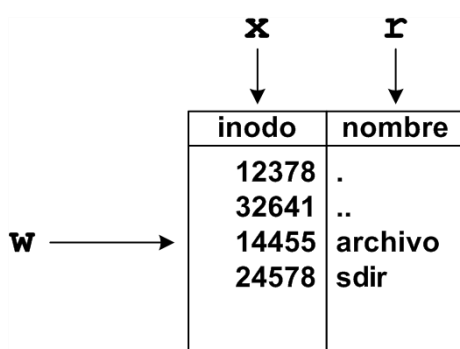
- tiene permiso para leer el archivo *notas*.
- tiene permiso para modificar el archivo *notas*.
- no tiene permiso para ejecutar el archivo *notas*.

Más exactamente, se distinguen los permisos Unix estándar según el tipo de archivo: archivo normal o directorio.

Permiso	Archivo	Directorio
r	Permiso de lectura del contenido del archivo.	Permiso para listar las entradas del directorio.
w	Permiso para modificar el contenido del archivo.	Permiso para modificar las entradas del directorio.
x	Permiso para ejecutar el archivo.	Permiso de acceso a las entradas del directorio.

Es relativamente simple comprender los permisos correspondientes cuando se otorgan a un archivo normal; esto no es tan evidente cuando se trata de un directorio.

Si consideramos los directorios como tablas que contienen, en una columna, los inodos y en otra, los nombres de archivo presentes en el directorio, es más fácil comprender los permisos estándar:



Capítulo 4

Edición de archivos de texto - Vi

Duración: 1 hora 30

Palabras clave

Editor, texto, vi, vim.

Objetivo

Al finalizar este capítulo, será capaz de editar archivos de texto con la ayuda del editor Vim disponible en Linux.

Estos ejercicios corresponden al capítulo 6 "Edición de archivos de texto - Vi" del libro "Linux - Principios básicos de uso del sistema 6ª edición" de la colección Recursos Informáticos de Ediciones ENI.

Material necesario

Estos ejercicios pueden realizarse en cualquier distribución Linux donde el lector posea una cuenta de usuario válida; la cuenta de usuario **tux** se utilizará en la corrección de los enunciados.

Requisitos previos

Para validar los requisitos previos necesarios, antes de iniciar la PT, responda las preguntas siguientes:

1. ¿En qué sistemas operativos se puede utilizar Vi?
 - a. En GNU/Linux.
 - b. En cualquier sistema Unix.
 - c. En los sistemas Microsoft Windows.
 - d. En Mac OS.
 - e. En todos los sistemas operativos enunciados anteriormente y algunos otros.
2. ¿Es cierto que Vim posee las mismas funcionalidades que Vi?
 - a. Si.
 - b. No.
3. ¿Cuáles son los tres modos de funcionamiento de Vi?

4. ¿Cuál es la secuencia de teclas que permite salir de Vi en todo momento sin salvar las modificaciones aportadas al archivo?
 - a. `:q`, luego [Entr]
 - b. `:q!`, luego [Entr]
 - c. [Esc], luego `:q`, luego [Entr]
 - d. [Esc], luego `:q!`, luego [Entr]

Solución pág. 109

Enunciado 4.1 Inicialización y salida de Vi

Duración estimada: 2 minutos

1. Conéctese a la tercera consola virtual texto (tty3) como el usuario `tux`.
2. Inicie el editor de texto Vi sin argumento. ¿Qué observa?
3. Salga de Vi.
4. Edite ahora el archivo `/etc/hosts`. ¿Qué significan los caracteres `~` situados al comienzo de línea?
5. Salga de Vi.

Pista para el enunciado 4.1

2. *A lo largo de estos ejercicios, podrá utilizar indiferentemente el comando `vi` o el comando `vim`.*

Solución pág. 110

Enunciado 4.2 Comandos de desplazamiento

Duración estimada: 10 minutos

1. Edite el archivo `/etc/services`.
2. Sin utilizar las teclas de dirección del teclado (flechas), desplace el cursor una línea hacia abajo.
3. Sin utilizar las teclas de dirección del teclado, desplace el cursor doce líneas hacia abajo.
4. Sin utilizar las teclas de dirección del teclado, desplace el cursor ocho líneas hacia arriba.
5. Sin utilizar las teclas de dirección del teclado, desplace el cursor dieciséis caracteres hacia la derecha.

6. Sin utilizar las teclas de dirección del teclado, desplace el cursor nueve caracteres hacia la izquierda.
7. Desplace el cursor al final de línea.
8. Desplace el cursor al principio de línea.
9. Desplace varias veces el cursor hasta el primer carácter de la palabra siguiente.
10. Desplace varias veces el cursor hasta el primer carácter de la palabra anterior.
11. Desplace varias veces el cursor hasta el último carácter de la palabra siguiente.
12. Desplace el cursor hasta la última línea del archivo.
13. Desplace el cursor hasta la primera línea del archivo.
14. Desplace el cursor hasta la línea 45 del archivo.
15. Desplace el cursor una página hacia arriba.
16. Desplace el cursor una página hacia abajo.

Solución pág. 113

Enunciado 4.3 Comandos de edición y de corrección

Duración estimada: 15 minutos

1. Desplace el cursor hasta el sexto carácter de la primera línea del archivo.
2. Inserte la cadena de caracteres "uno" delante del cursor en la línea actual.
3. Sin utilizar las teclas de dirección del teclado, desplácese tres caracteres hacia la derecha e inserte la cadena de caracteres "dos" después del cursor.
4. Inserte la cadena de caracteres "tres" al principio de la línea.
5. Inserte la cadena de caracteres "cuatro" al final de la línea.
6. Sin utilizar las teclas de dirección del teclado, desplace el cursor seis caracteres hacia la izquierda.
7. Suprima el carácter bajo el cursor.
8. Suprima el carácter situado delante del cursor.
9. Suprima los cinco caracteres precedentes al cursor.
10. Borre todos los caracteres situados desde el cursor hasta el final de línea.
11. Sin utilizar las teclas de dirección del teclado, desplace el cursor seis caracteres hacia la izquierda.
12. Borre todos los caracteres situados desde el inicio de línea hasta el cursor.

13. Pegue al final de línea el texto que acaba de “cortar”.
14. Borre la totalidad de la línea.
15. Copie la última línea del archivo y péguela al inicio del archivo.
16. Anule el último comando. ¿Es posible anular también los comandos anteriores?
17. Borre las cuatro primeras líneas del archivo.
18. Borre las cuatro líneas siguientes.

Pistas para el enunciado 4.3

2. *Debe entrar en modo "edición" de Vi.*
3. *No olvide volver al modo "comando" antes de desplazar el cursor, después entre de nuevo en modo "edición".*
4. *Utilice el comando que permite pasar a modo "edición" insertando directamente al inicio de línea.*
5. *Utilice el comando que permite pasar a modo "edición" insertando directamente al final de línea.*
10. *Utilice el comando de borrado seguido del comando de desplazamiento adecuado.*
12. *Utilice el comando de borrado seguido del comando de desplazamiento adecuado.*
18. *Utilice el comando de repetición.*

Solución pág. 114

Enunciado 4.4 Comandos globales

Duración estimada: 10 minutos

1. Intente guardar las modificaciones que ha realizado en el archivo. ¿Es posible?
2. Guarde el trabajo en su directorio personal con el nombre de archivo *services2*.
3. Suprima la primera línea del archivo, después guárdelo y salga de Vi.
4. Abra de nuevo el archivo */etc/services* y busque la cadena de caracteres "tcp".
5. Desplace el cursor de ocurrencia en ocurrencia de la cadena de caracteres "tcp".
6. Desplace siempre el cursor de ocurrencia en ocurrencia de la cadena de caracteres "tcp", pero esta vez, en el sentido inverso de la búsqueda.
7. Reemplace todas las cadenas de caracteres "udp" del archivo por "UDP".

Pista para el enunciado 4.4

2. *Utilice una ruta personal para especificar el nuevo nombre del archivo.*

Solución pág. 115

Enunciado 4.5 Opciones del editor

Duración estimada: 10 minutos

1. Visualice todas las opciones del editor.
2. Visualice los números de líneas.
3. Defina el numero de espacios a 2 para representar una tabulación.
4. Salga de Vi sin guardar y abra de nuevo el archivo */etc/services*.
5. ¿Las opciones son siempre efectivas? ¿Cómo hacer para que estas modificaciones sean permanentes?

Pistas para el enunciado 4.5

2. *Active la opción adecuada..*
3. *Modifique el valor de la opción adecuada.*

Solución pág. 116

Enunciado 4.6 Para ir más lejos

Duración estimada: 40 minutos

1. Si el comando está presente en su sistema, inicie **vimtutor** y siga los ejercicios propuestos.
2. Busque y consulte la documentación existente que trate del editor de texto Vim.
3. Busque y utilice otros editores de texto de su distribución GNU/Linux.

Pista para el enunciado 4.6

2. *La documentación Linux fue abordada en el capítulo Documentación de esta obra.*

Solución pág. 117

