
Unidad 3: Gestión de Usuarios y Grupos Permisos y Procesos

**Curso de Administración de Servidores GNU/Linux
Centro de Formación Permanente
Universidad de Sevilla**

Jorge Juan <jjchico@dte.us.es>, Enrique Ostúa <ostua@dte.us.es>, 2010-2014

Usted es libre de copiar, distribuir y comunicar públicamente la obra y de hacer obras derivadas bajo las condiciones de la licencia Attribution-Share alike de Creative Commons.

Puede consultar el texto completo de la licencia en <http://creativecommons.org/licenses/by-sa/3.0/>

Contenidos

- Usuarios y grupos
- Gestión de usuarios y grupos
- Usuarios y grupos especiales
- Permisos
- Límites
- Procesos

Usuarios y grupos

- GNU/Linux es un sistema:
 - Multiusuario: varias personas pueden trabajar con el sistema, que proporciona mecanismos de protección de datos de cada usuario.
 - Multitarea: varios procesos (de varios usuarios) pueden ejecutarse a la vez
 - De tiempo compartido: los procesos se ejecutan todos todo el tiempo
- Cuenta de usuario: datos e información de cada usuario. Cada usuario posee un nombre de usuario y una clave para acceder al sistema.
- Grupo de usuarios: cada usuario puede pertenecer a uno o varios grupos
- Permisos: sistema de protección de datos de un usuario respecto de otros.

Administración de Servidores GNU/Linux

3

Usuarios y grupos

- Características de las cuentas de Usuario:
 - Nombre de usuario (“username”)
 - Identifica al usuario dentro del sistema (cada usuario lleva asociado un número de identificación)
 - Clave (password)
 - para acceder al sistema
 - Grupo principal del usuario
 - En Ubuntu “por defecto” se crea un grupo de mismo nombre que el “username”
 - Carpeta personal
 - Donde el usuario guarda sus propias carpetas y archivos (por defecto en /home/username)
 - Otros grupos a los que pertenezca el usuario

Administración de Servidores GNU/Linux

4

Contenidos

- Usuarios y grupos
- **Gestión de usuarios y grupos**
- Usuarios y grupos especiales
- Permisos
- Límites
- Procesos

Gestión de usuarios en entorno gráfico

- Configuración del sistema → Cuentas de usuario
 - Configuración del usuario propio
 - Configuración básica de otros usuarios (administrador)
 - Añadir/eliminar usuarios

Gestión de usuarios y grupos en interfaz de comandos

- Añadir usuarios:
 - # adduser <username>
- Asignar un usuario a un grupo
 - # adduser <username> <groupname>
- Borrar usuarios:
 - # userdel <username>
- Añadir grupos:
 - # addgroup <groupname>
- Borrar grupos:
 - # groupdel <groupname>

Gestión de usuarios y grupos en interfaz de comandos

- Modificando propiedades de los usuarios y grupos (usermod, groupmod). Ejemplos:
 - # usermod -d /home/profes/pepe -m
cambia la carpeta del usuario pepe a /home/profes/pepe. -m hace que se mueva todo el contenido.
 - # usermod -l joseg pepe:
cambia nombre de usuario pepe por joseg.
 - # usermod -g profes pepe
cambia grupo principal del usuario pepe a profes.
 - #groupmod -n profesores profes
cambia el nombre del grupo profesores a profes

Cambios de propietario/grupo

- Desde el administrador de archivos
 - Menú contextual -> Propiedades -> Permisos
- Cambiar propietarios de archivos/carpetas
 - `chown [-R] <usuario> <ruta>`
 - `chown [-R] <usuario>:<grupo> <ruta>`
- Cambiar grupo de un archivo/carpeta
 - `chgrp [-R] <grupo> <ruta>`

Gestión de usuarios y grupos. Archivos de configuración

- Opciones por defecto al crear usuarios:
 - `/etc/adduser.conf`
- Contenido inicial de las nuevas cuentas de usuarios:
 - `/etc/skel`
- Ficheros con información sobre usuarios y grupos.
Normalmente no se editan “a mano”
 - `/etc/passwd`: Usuarios, ID, grupo principal, carpeta inicio, shell, etc.
 - `/etc/shadow`: continen los passwords encriptados de los usuarios.
 - `/etc/group`: Grupos, miembros de cada grupo.
 - Se usa “`vipw`” y “`vigr`” para editarlos con seguridad

Contenidos

- Usuarios y grupos
- Gestión de usuarios y grupos
- **Usuarios y grupos especiales**
- Permisos
- Límites
- Procesos

Usuarios y grupos especiales

- Usuarios y/o grupos empleados por el sistema para realizar tareas concretas con privilegios restringidos
 - cupsys: administrador de impresoras
 - mail: gestor de correo
 - www-data: servidor web
 - ...
- Grupos para el control de privilegios
 - cdrom: control de las unidades de cd o dvd
 - floppy: control de las unidades de diskette
 - audio: control de los dispositivos de audio
 - admin: dota de permisos de administrador del sistema a sus miembros (a través de sudo)

Usuarios y grupos especiales. Superusuario

- Superusuario = administrador = root
- Superusuario -> privilegios ilimitados
- Los sistemas GNU/Linux se configuran para realizar el mínimo de tareas empleando el superusuario
- Convertirse en otro usuario
 - `$ su [-] <nuevo_usuario>`
- Ejecutar un comando como superusuario (sólo usuarios grupo "sudo")
 - `$ sudo ...`
- Convertirse en root permanentemente
 - `$ sudo -s`
 - `$ sudo su`

Contenidos

- Usuarios y grupos
- Gestión de usuarios y grupos
- Usuarios y grupos especiales
- **Permisos**
- Límites
- Procesos

Permisos

- Cada archivo/carpeta tiene un propietario y está asignado a un grupo (al que normalmente pertenece el propietario)
- Cada archivo/carpeta posee tres conjuntos de permisos que controlan el acceso de:
 - El propietario (user)
 - Miembros del grupo al que pertenece el archivo (group)
 - Otros usuarios (other)

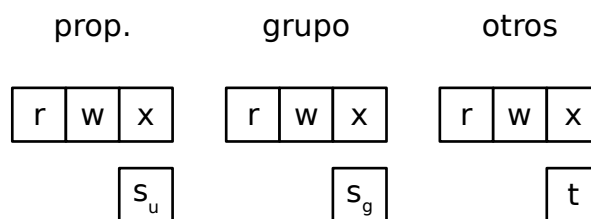
Permisos básicos

| | Lectura/listado (r) | Escritura (w) | Ejecución/ acceso (x) |
|----------------|--|---|---|
| Archivo | se puede leer el contenido del archivo | se puede cambiar el contenido del archivo | se puede ejecutar el archivo como un programa |
| Carpeta | se puede listar el contenido de la carpeta | se pueden crear archivos en la carpeta | se puede entrar en la carpeta |

Permisos especiales

| | Set-uid (su) | Set-gid (sg) | Sticky (t) |
|-------------------------|---|---|---|
| Archivo (ejecutable) | ejecución con privilegios del propietario | ejecución con privilegios del grupo (poco útil) | hacer residente (no se usa) |
| Carpeta | (no se usa) | crea nuevos archivos con el grupo de la carpeta padre | permiso de escritura sólo para propietario del archivo (/tmp) |

Permisos. Notación



- lectura: r
- escritura: w
- ejecución: x
- set-uid: s
- set-gid: S
- sticky: t
- X -> rwx
- S -> sólo "s"
- T -> sólo "t"

Permisos

- Gestionar permisos. Administrador de archivos
 - Menú contextual -> Propiedades -> Permisos
- Gestionar permisos
 - Comando chmod
- Cambiar propietarios de archivos/carpetas
 - # chown [-R] <usuario> <ruta>
 - # chown [-R] <usuario>:<grupo> <ruta>
- Cambiar grupo de un archivo/carpeta
 - # chgrp [-R] <grupo> <ruta>

Permisos. chmod

```
chmod [-R] u+xXs <rutas>
          g+Xs
          o-
          a=
          r
          w
          x
          X
          s
          t
          u
          g
          o
```

Permisos. chmod

| especial | prop. | grupo | otros | | | | | | | | | | | | |
|---|----------------|----------------|-------|---|---|---|---|---|---|---|---|---|---|---|---|
| <table border="1"><tr><td>s_u</td><td>s_g</td><td>t</td></tr></table> | s _u | s _g | t | <table border="1"><tr><td>r</td><td>w</td><td>x</td></tr></table> | r | w | x | <table border="1"><tr><td>r</td><td>w</td><td>x</td></tr></table> | r | w | x | <table border="1"><tr><td>r</td><td>w</td><td>x</td></tr></table> | r | w | x |
| s _u | s _g | t | | | | | | | | | | | | | |
| r | w | x | | | | | | | | | | | | | |
| r | w | x | | | | | | | | | | | | | |
| r | w | x | | | | | | | | | | | | | |
| <table border="1"><tr><td>0</td><td>0</td><td>0</td></tr></table> | 0 | 0 | 0 | <table border="1"><tr><td>1</td><td>1</td><td>0</td></tr></table> | 1 | 1 | 0 | <table border="1"><tr><td>1</td><td>0</td><td>0</td></tr></table> | 1 | 0 | 0 | <table border="1"><tr><td>0</td><td>0</td><td>0</td></tr></table> | 0 | 0 | 0 |
| 0 | 0 | 0 | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | |
| (0) | 6 | 4 | 0 | | | | | | | | | | | | |
| <table border="1"><tr><td>0</td><td>1</td><td>0</td></tr></table> | 0 | 1 | 0 | <table border="1"><tr><td>1</td><td>1</td><td>1</td></tr></table> | 1 | 1 | 1 | <table border="1"><tr><td>1</td><td>0</td><td>1</td></tr></table> | 1 | 0 | 1 | <table border="1"><tr><td>1</td><td>0</td><td>1</td></tr></table> | 1 | 0 | 1 |
| 0 | 1 | 0 | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | |
| 2 | 7 | 5 | 5 | | | | | | | | | | | | |

chmod [-R] <código octal> <rutas>

Permisos

- Ejemplos

```
$ chmod g+r carta.txt
$ chmod a+x programa
$ chmod go-rw proyecto.odt
$ chmod g+s mi_carpeta
$ chmod -R a+rX carpeta_web
$ chmod g=u carta.txt
$ chmod 775 carpeta_web
$ chmod 2750 proyecto
```

Permisos

- La combinación de usuarios, grupos y permisos permite establecer múltiples políticas de privacidad.
- Ejemplos:
 - Contenido privado del usuario: quitar todos los permisos a grupo y otros
 - Compartir contenido privado entre varios usuarios
 - Crear un grupo con todos los usuarios implicados
 - Asociar contenidos a ese grupo
 - Establecer permisos de grupo: ej: lectura+escritura
 - Quitar todos los permisos para otros
 - Contenido público que sólo algunos usuarios pueden modificar
 - Como anterior, con permiso de lectura/acceso para otros

Listas de control de acceso

- Sistema de permisos más avanzado
- No activo de forma predeterminada en Ubuntu (opción de montaje “acl”)
- Mayor granularidad que el sistema tradicional:
 - Permisos para usuarios distintos del propietario.
 - Permisos para grupos distintos del grupo del archivo.
 - Permisos “por defecto”
- Comandos:
 - getfacl
 - setfacl

Contenidos

- Usuarios y grupos
- Gestión de usuarios y grupos
- Usuarios y grupos especiales
- Permisos
- **Límites**
- Procesos

Límites y restricciones

- Existen diversos sistemas para imponer límites y restricciones a los usuarios y sus procesos (configuración en **/etc/security**):
 - limits.conf, limits.d
 - Límites a recursos usados por los usuarios: tamaño máximo de archivo, número de archivos abiertos, cantidad de memoria asignable, tiempo de CPU, número de procesos, etc.
 - access.conf, time.conf
 - Límite sobre acceso de usuarios al sistema (lugar, hora, etc.)
 - Etc.

Contenidos

- Usuarios y grupos
- Gestión de usuarios y grupos
- Usuarios y grupos especiales
- Permisos
- Límites
- **Procesos**

Control de procesos

- Proceso (*process*) es cada una de las tareas (*jobs*) que se realiza en el sistema.
- Cada vez que se ejecuta una aplicación o programa se inicia un nuevo proceso.
- Control de procesos
 - Monitor del sistema (comando: `gnome-system-monitor`)
 - Comandos de terminal:
 - `ps`
 - `jobs`
 - `kill`
 - `killall`
 - ...

Control de procesos

- Monitor del sistema: Procesos
 - **Nombre** del proceso y el **PID** (identificar único)
 - Estado: Durmiendo, ejecutándose, detenido...
 - Memoria RAM o de intercambio que ocupa
 - Porcentaje de CPU que esta utilizando un proceso
 - Prioridad: Prioridad con que se ejecuta un proceso.
 - Puede variar desde 0 hasta 20. A mayor valor menor prioridad. Por defecto tienen prioridad 0.
 - Un usuario sólo puede cambiar la prioridad de sus propios procesos.
 - El administrador puede cambiar la prioridad de cualquier proceso entre -20 y 19.

Control de procesos

- Acciones sobre los procesos:
 - Detener Proceso: Para la ejecución de un proceso sin eliminarlo (es posible reanudar su ejecución con continuar)
 - Finalizar proceso o Matar proceso: Elimina este proceso.
 - Cambiar la prioridad: Los valores negativos sólo los puede ejecutar el administrador.
 - Ocultar procesos: No muestra el proceso seleccionado en la lista de procesos.

Control de procesos

- Existen comandos para la interfaz de comando que permiten visualizar, parar o matar los diferentes procesos:
 - ps o top: para visualizar los procesos que se ejecutan
 - kill, killall: para matar procesos
- Ejemplos:
 - \$ ps -u *usuario* ; \$ ps -fu *usuario*
 - \$ pstree (muestra las relaciones: padres e hijos)
 - \$ top (se sale con 'q', ayuda con '?')
 - \$ ps -feL (-L muestra los threads de cada proceso)
 - \$ kill 20258 ('mata' ese PID)
 - \$ killall nautilus ('mata' por nombre)

Control de procesos

- Desde shell podemos ejecutar en modo “*foreground*” (interactivo, el habitual) o “*background*” (no-interactivo, se desvinculan la entrada y salida de la shell)
 - \$ comando → lo lanza interactivo “fore”
 - \$ comando & → lo lanza no-interactivo “back”
 - En “fore”: Ctrl-z detiene el proceso; Ctrl-c lo mata.
 - Detenido: \$ bg continúa en “back” y \$ fg en “fore”
 - En “back”: \$ fg lo devuelve a “fore”
- Los procesos en “*back*” pasan a ser 'trabajos' de esa terminal, se pueden listar con \$ jobs y se numeran %1, %2, etc...
 - \$ jobs \$ fg %1
 - \$ bg %2 \$ kill %3