
Usuarios, Grupos y Permisos en GNU/Linux

Nota de Copyright

© 2005 Diego Chaparro. Algunos derechos reservados.

Este trabajo se distribuye bajo la licencia Creative Commons Attribution-ShareAlike. Para obtener la licencia completa, véase <http://creativecommons.org/licenses/by-sa/2.1/es>

Usuario root

- ◆ Cuando instalamos un sistema el único usuario que se crea el es el root
- ◆ root es el administrador del sistema, puede hacer cualquier cosa:
 - ◆ Puede acceder a cualquier dispositivo
 - ◆ Puede borrar cualquier parte del sistema
- ◆ Por eso el usuario root **solo** debe ser usado para realizar tareas de administración
- ◆ El uid del root es 0, pero el nombre root puede ser cambiado a otro nombre, siempre conservando el uid

SU

- ◆ Permite realizar cambios de usuario con el que se está “logueado”
- ◆ Se puede usar paraa:
 - ◆ Un usuario normal tiene que realizar algo como root
 - ◆ El usuario root necesita hacer algo como un usuario normal
- ◆ Si queremos que todas las variables de entorno se cambien al usuario que cambiamos hay que usar la opción -. Ejemplo:

su – usuario

Gestionar usuarios

◆ **/etc/passwd**

- ◆ Toda la información relativa a las cuentas de usuario está especificado en este fichero, con el siguiente formato:

Nombre de usuario:

- ◆ Debe ser único en el sistema.
- ◆ Puede contener 8 caracteres o menos, no se deben usar caracteres no alfanuméricos
- ◆ Para sistemas grandes hay que ver métodos para asignar nombres, (Ej: primera letra del nombre + apellido, ...)

Contraseña:

- ◆ Las contraseñas se almacenan cifradas
- ◆ Si se pone un * el usuario no podrá hacer login en el sistema, pero si podrá realizar otras acciones como ejecutar comandos remotos
- ◆ Si se deja el campo vacío, al hacer login no pedirá password

Gestionar usuarios

◆ **/etc/passwd**

Identificador de usuario:

- ◆ El uid es el identificador del usuario para realizar cualquier acción (ejecutar procesos, dueño de ficheros, ...)
- ◆ Los rangos de UID definen el tipo de cuenta:
 - ◆ Los menores de 100 se reservan normalmente para cuentas del sistema
 - ◆ A los usuarios se le asignan cuentas con UID mayor de 100 o 1000, dependiendo de la distribución

Identificador de grupo:

- ◆ Los grupos se usan para organizar a los usuarios
- ◆ Los rangos de GID se definen como:
 - ◆ Del 1 al 49 se suelen usar como grupos del sistema
 - ◆ A partir del 50 suelen ser para grupos de usuario

Gestionar usuarios

◆ **/etc/passwd**

Comentarios:

- ◆ Este campo se usa para almacenar información general sobre los usuarios
- ◆ Se suele almacenar el nombre, departamento, teléfono, ...

Directorio HOME:

- ◆ Se especifica el directorio donde se situará el usuario al hacer login en el sistema

Intérprete de comandos:

- ◆ Determina el comando que se ejecutará cuando el proceso de login termine
- ◆ La mayoría de las veces ahí se encontrará un intérprete de comandos
- ◆ Si se deja en blanco se ejecutará la shell por defecto

Crear usuarios

- ◆ Al menos debe estar definido el nombre de usuario y el directorio HOME
- ◆ Existen varios métodos para añadir usuarios:
 - ◆ **Manual:**
 - ◆ Editando el fichero `/etc/passwd`
 - ◆ Cambiando la password
 - ◆ Crear el directorio de usuario y copiando los archivos de configuración
 - ◆ Añadir el grupo si es necesario
 - ◆ **Automática:**
 - ◆ `useradd`. Después habría que cambiar la password porque pone un `!!` en ese campo. Tampoco se crea el directorio por defecto a no ser que no se especifique la opción `-m`
 - ◆ `adduser`: Más avanzado, va preguntando todos los datos.

Crear usuarios

◆ **Contraseñas**

- ◆ Deben tener al menos 6 caracteres
- ◆ No deben usar palabras de diccionarios
- ◆ Deben contener números, mayúsculas, minúsculas, ...

◆ **passwd**

- ◆ Permite cambiar la contraseña de un usuario
- ◆ El usuario root puede cambiar la contraseña de cualquier usuario

Crear usuarios

◆ **chpasswd**

- ◆ Actualiza las contraseñas del sistema a partir de los datos de un fichero con el siguiente formato:

nombre:contraseña

nombre:contraseña

shadow passwords

- ◆ Cuando el campo contraseña en `/etc/passwd` es `x`, el sistema está usando un archivo de shadow passwords
- ◆ Este fichero contiene el nombre de usuario y la contraseña cifrada, y otros campos:
 - ◆ Nombre de usuario
 - ◆ Contraseña cifrada
 - ◆ Número de días que han pasado desde 1970 y el día cuando se cambió la password por última vez
 - ◆ Mínimo tiempo que debe esperar para poder cambiar la contraseña
 - ◆ Cuantos días puede estar sin cambiar la contraseña
 - ◆ Cuantos días se le debe avisar al usuario para que cambie su password antes de que caduque
 - ◆ Cuantos días esperar a deshabilitar la cuenta desde que le caduque
 - ◆ Número de días en los que caducará la password
- ◆ El fichero `/etc/shadow` solo puede ser leído por el usuario root

shadow passwords

◆ Shadow passwords

◆ pwconv

- ◆ Convierte el sistema de contraseñas normales en shadow passwords
- ◆ Crea el fichero `/etc/shadow` y mete una x en el campo contraseña de `/etc/passwd`

◆ pwunconv

- ◆ Convierte un sistema con shadow passwords en uno sin ellas
- ◆ Borra el fichero `/etc/shadow` y pone las contraseñas en `/etc/passwd`

usermod

- ◆ Permite modificar los campos de `/etc/passwd`
- ◆ Sintaxis: `usermod [opcion] usuario`
- ◆ Opciones:
 - l usuario Cambia el nombre de usuario
 - u n Cambia el UID
 - g n Cambia el GID
 - c "information" Cambia la información
 - d path Cambia el HOME de usuario
 - s path Cambia el intérprete de comandos
- ◆ No se puede cambiar el nombre de usuario si está logeado

/etc/login.defs

- ◆ Define ciertos parámetros de los logins y las cuentas del sistema
- ◆ Define como tienen que ser las passwords:
 - ◆ Tamaño
 - ◆ Tiempo para caducar
 - ◆ Tiempo que será preguntado el usuario antes de que le caduque la password
- ◆ Inicio de rangos de UID y GID de usuarios
- ◆ Directorio del correo de usuarios

finger

- ◆ Información sobre un usuario del sistema:
 - ◆ Nombre de usuario
 - ◆ Información de /etc/passwd
 - ◆ Directorio HOME
 - ◆ Intérprete de comandos
 - ◆ Los últimos accesos
 - ◆ ...

chsh

- ◆ Permite cambiar a un usuario la shell que tiene
 - ◆ Ejemplo:
`chsh -s /bin/csh usuario`

Usuarios especiales

- ◆ Algunas cuentas de las que aparecen en /etc/passwd son cuentas del sistema:
 - ◆ root Administrador del sistema
 - ◆ daemon Posee y mantiene los permisos de los procesos del sistema
 - ◆ bin Posee los ejecutables
 - ◆ sys Posee ejecutables
 - ◆ adm Dueño de los ficheros de log
 - ◆ mail
 - ◆ ftp
 - ◆

Borrar usuarios

- ◆ Manual:
 - ◆ Borrar la línea de usuario de `/etc/passwd`
 - ◆ Borrar el directorio de usuario
 - ◆ Buscar y borrar ficheros del usuario que estén fuera de su directorio de usuario
 - ◆ Borrar el correo del usuario y los posibles alias
 - ◆ Borrar las posibles tareas planificadas que tenga el usuario
- ◆ Automática:
 - ◆ `userdel`: No borra el directorio HOME
 - ◆ `deluser`: Más avanzado
- ◆ Deshabilitar una cuenta de usuario:
 - ◆ Poner un `*` delante de la contraseña en `/etc/passwd`

Gestionar grupos

- ◆ Los grupos son usados para organizar los usuarios y para otorgar permisos a los ficheros
- ◆ Ejemplos de grupos:
 - ◆ Grupos para acceder a dispositivos
 - ◆ Grupos para acceder a partes del sistema de ficheros
- ◆ **/etc/group**
 - ◆ Los campos de este fichero son:
 - ◆ Nombre del grupo
 - ◆ Contraseña
 - ◆ Identificador de grupo
 - ◆ Miembros del grupo separados por comas

Gestionar grupos

- ◆ **newgrp**

- ◆ Especifica cuál es el grupo por defecto de un usuario
- ◆ El grupo por defecto se usa por ejemplo para especificar el grupo de un nuevo fichero creado

- ◆ **Crear grupos:**

- ◆ Manual:
 - ◆ Editando /etc/group
- ◆ Automática:
 - ◆ groupadd
 - ◆ addgroup

Gestionar grupos

◆ Añadir usuarios a grupo

◆ gpasswd:

◆ Añade un usuario a un grupo

◆ Ejemplo: `gpasswd -a usuario grupo`

◆ adduser:

◆ Ejemplo: `adduser usuario grupo`

◆ Con `gpasswd` se puede establecer la contraseña de grupo. Si un grupo tiene contraseña, un usuario puede unirse al grupo dando la contraseña.

◆ Opciones de `gpasswd`:

-R Previene el uso de `newgrp` para unirse al grupo

-a usuario Añade un usuario al grupo

-d usuario Quita a un usuario de un grupo

groupmod

- ◆ Permite modificar los datos de `/etc/group`
- ◆ Sintaxis: `groupmod [opcion] grupo`
- ◆ Opciones:
 - n nombre Cambia el nombre de grupo
 - g gid Cambia el GID del grupo
 - o Especifica un GID no único

Borrar grupos

- ◆ Manual:
 - ◆ Borrar la línea de `/etc/group`
 - ◆ Chequear `/etc/passwd` y verificar que ningún usuario tiene ese grupo
 - ◆ Verificar que no hay ficheros ni directorios con ese grupo
- ◆ Automática:
 - ◆ `groupdel`
 - ◆ `delgroup`

Gestionar grupos

◆ Grupos del sistema

- ◆ root Dueño de la mayoría de ficheros del sistema
- ◆ daemon Dueño del correo, impresora y otro software del sistema y directorios
- ◆ kmem Gestiona el acceso directo a la memoria del kernel
- ◆ sys Dueño de ficheros del sistema, ficheros de intercambio, e imágenes de memoria
- ◆ nobody Dueño de software sin permisos especiales
- ◆ tty Ficheros de dispositivos que controlan las terminales
- ◆ users Usuarios del sistema

◆ groups

- ◆ Muestra los grupos a los que pertenece un usuario

Variables de entorno

◆ **/etc/profile**

- ◆ En este fichero están definidas las variables de entorno por defecto para todos los usuarios, como por ejemplo:

- ◆ PATH
- ◆ PS1

◆ **Configuración de usuario**

- ◆ La shell mira en `.bash_profile`, `.bash_login` y `.profile` para ejecutarlos en ese orden
- ◆ Si hay dos variables iguales en estos ficheros, el valor que persiste es el último
- ◆ Si se inicia una shell sin login (al abrir un terminal por ejemplo) se ejecuta el `.bashrc`
- ◆ Cuando se cierra la sesión se ejecuta `.bash_logout`

Variables de entorno

- ◆ Para ver los valores de una variable:
 - ◆ echo \$VARIABLE
- ◆ **env**
 - ◆ env [NAME=VALUE] [comando]
 - ◆ Podemos ejecutar un comando con las variables de entorno especificadas
 - ◆ Si no pasamos argumentos vemos las variables de entorno del usuario actual

Alias

- ◆ Establece un alias para un comando
 - ◆ Ejemplo:
 - ◆ `alias rm="rm -i"`
 - ◆ `alias ll="ls -l"`

Variables de entorno

- ◆ Volver a cargar un archivo de configuración:
 - ◆ `source .bashrc`
 - ◆ `..bashrc`
- ◆ **PATH**
 - ◆ Especifica donde se buscan los comandos que se ejecutan
 - ◆ Para añadir al PATH:
 - ◆ `PATH=$PATH:/sbin`
 - ◆ `export PATH`
 - ◆ No se debe especificar el directorio actual en el path, problema de seguridad

Variables de entorno

◆ PROMPT

◆ Variable PS1

◆ Opciones:

\h nombre de máquina

\u nombre de usuario

\w directorio actual

◆ Ejemplo: PS1=\u@\h:\w\$

Permisos

◆ Dueño y grupo

- ◆ Todos los ficheros del sistema poseen un dueño y un grupo
- ◆ El dueño suele ser el que ha creado el fichero, y el grupo suele ser el grupo por defecto de ese usuario

◆ chown

- ◆ Cambia el dueño de un fichero
- ◆ Sintaxis: `chown [opcion] usuario archivo`
- ◆ Opciones:
 - ◆ `-c` Muestra información de todos los cambios
 - ◆ `-f` No muestra mensajes de error
 - ◆ `-R` Recursivamente

Permisos

◆ Cambiar el grupo

- ◆ Se puede hacer con `chown`:
 - ◆ `chown usuario[:l.]grupo fichero(s)`
 - ◆ `chown [:l.]grupo fichero(s)`
- ◆ Con **chgrp**:
 - ◆ `chgrp grupo fichero(s)`

Permisos

- ◆ El bloque de permisos consta de 10 caracteres:
 - ◆ Tipo de fichero
 - ◆ Permisos del dueño (3 caracteres)
 - ◆ Permisos del grupo (3 caracteres)
 - ◆ Permisos para los demás (3 caracteres)
- ◆ Los permisos básicos de un fichero se representan mediante:
 - ◆ r (read)
 - ◆ w (write)
 - ◆ x (execute)

Permisos

◆ Acceso a ficheros

- ◆ r (read):
 - ◆ Permite ver el contenido del fichero, incluyendo la edición con un editor
- ◆ w (write):
 - ◆ Permite modificar el contenido del fichero
 - ◆ No podemos editarlo si no tenemos permiso de lectura
 - ◆ No permite borrar el fichero
- ◆ x (execute):
 - ◆ Permite ejecutar el fichero

Permisos

◆ Acceso a directorios

◆ r:

- ◆ Se puede listar el contenido del directorio
- ◆ No significa que podamos entrar en ese directorio

◆ w:

- ◆ Permite crear ficheros y directorios dentro
- ◆ También permite borrar ficheros que haya dentro, incluso aunque no tengamos permisos sobre ellos

◆ x:

- ◆ Permite cambiarse a ese directorio
- ◆ Si no tenemos r no podremos ver el contenido

chmod

◆ **chmod (símbolos)**

◆ Solo el propietario o el root puede cambiar los permisos de un fichero o directorio

◆ Sintaxis: *chmod* <quien> <cambio> <permisos> [ficheros]

◆ quien:

u	propietario	g	grupo
o	otros		

Si no se especifica, lo cambia a todos

◆ cambio:

-	Quita permisos	+	Dá permiso
----------	----------------	----------	------------

◆ permisos:

r	w	x
----------	----------	----------

chmod

◆ **chmod (símbolos)**

◆ Ejemplos:

◆ `chmod g+w fichero`

Añade permiso de escritura para el grupo

◆ `chmod g=w fichero`

El grupo solo tiene permiso de escritura

◆ `chmod ug+x fichero`

Añade permiso de ejecución para propietario y grupo

◆ `chmod ug=x fichero`

Configura permiso de ejecución para dueño y grupo

◆ `chmod +rwx fich*`

Dá permiso de lectura, escritura y ejecución a dueño, grupo y otros de los ficheros que cumplen ese patrón

chmod

◆ **chmod (símbolos)**

◆ Ejemplos:

◆ `chmod o=g fichero`

Pone los permisos de otros iguales a los del grupo

◆ Opciones:

-R recursivo

-c Muestra los nombres de fichero de los que se han cambiado los permisos

-f No muestra mensajes de error

chmod

◆ **chmod (numeros)**

- ◆ En realidad, los permisos se identifican mediante 3 octetos
- ◆ Cada octeto representa los permisos de lectura, escritura y ejecución en binario:
 - ◆ 111 -> 7 (rwx)
 - ◆ 110 -> 6 (rw-)
 - ◆ 100 -> 4 (r--)
 - ◆ 010 -> 2 (-w-)
 - ◆ 001 -> 1 (--x)
- ◆ Al chmod se le pueden pasar estos número para cada uno de los bloques de permisos:
 - ◆ `chmod 764 fichero` (Permisos rwxrw-r--)

Permisos especiales

◆ SUID

- ◆ Permite ejecutar un fichero, y se ejecuta como si el que lo ejecuta fuera el dueño del fichero

◆ Ejemplo:

- ◆ `chmod o+s fichero`
- ◆ `chmod u+s fichero`

◆ GUID

- ◆ Permite ejecutar el fichero, y se ejecuta como si el grupo del que lo ejecuta fuera el grupo del fichero

- ◆ `chmod g+s fichero`

Permisos especiales

- ◆ También se puede cambiar mediante cuatro dígitos, el primero indica el tipo de permiso especial:
 - ◆ 4 Pone UID cuando ejecuta el fichero
 - ◆ 2 Pone GID cuando ejecuta el fichero
 - ◆ 1 Sticky bit
- ◆ Ejemplos:
 - ◆ `chmod 4755 fichero`
 - ◆ `chmod 2755 fichero`

Permisos especiales

- ◆ Sticky bit
 - ◆ Se suele usar para directorios con contenido que cualquiera puede modificar y añadir ficheros, pero queremos que solo el dueño pueda borrar ficheros
 - ◆ En lugar de tener drwxrwxrwx tenemos drwxrwxrwt
 - ◆ Ejemplo:
 - ◆ `chmod u+t directorio`
 - ◆ `chmod 1777 directorio`

Permisos por defecto

- ◆ La máscara de usuario (umask) define los permisos de los nuevos ficheros y directorios
- ◆ Para ficheros se hace un XOR (or exclusivo) de la máscara con 666. Por seguridad se quitan los permisos de ejecución.
- ◆ Para directorios se hace un XOR (or exclusivo) de la máscara con 777.
- ◆ Podemos ver la que tenemos con el comando **umask**

Permisos por defecto

- ◆ Cada dígito corresponde a dueño, grupo y otros, y puede ser:
 - ◆ 0 rw para ficheros. rwx para directorios
 - ◆ 1 rw para ficheros y directorios
 - ◆ 2 r para ficheros. rx para directorios
 - ◆ 3 r para ficheros y directorios
 - ◆ 4 w para ficheros. wx para directorios
 - ◆ 5 w para ficheros y directorios
 - ◆ 6 x para ficheros y directorios
 - ◆ 7 sin permisos
- ◆ Se pueden cambiar con el comando `umask`: `umask 027`
- ◆ El valor por defecto se define por ejemplo en `.bash_profile`