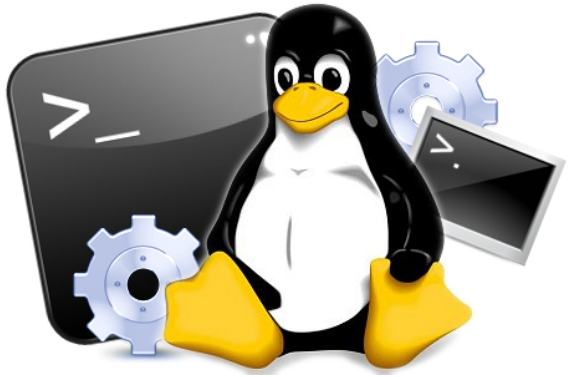
UT12.1: Administración de Linux: usuarios y permisos



FONDO SOCIAL EUROPEO

Javier Traseira

Usuarios y grupos

Linux como sabemos es un sistema operativo **multiusuario**, característica que permite por tanto ser utilizado por múltiples usuarios al mismo tiempo, compartiendo procesador, memoria, almacenamiento, aplicaciones y periféricos.

Para que múltiples usuarios puedan utilizar el sistema de forma segura y ordenada, es necesario que el sistema disponga de mecanismos de administración y seguridad para la protección de los datos de cada uno de los usuarios, así como herramientas que permitan su correcto funcionamiento.

Los usuarios del sistema pueden acceder a este de manera **local** y **remota**, siendo esta última manera la más común en este tipo de sistemas.

Usuarios y grupos locales

Cada usuario se identifica por un User ID o **UID** así como por su Group ID o **GID** asociado, que por defecto tiene su mismo nombre.

En general, a los usuarios generados se les asigna un UID igual o superior a *`1000`*. Los comprendidos entre *`0-100`* quedan reservados a usuarios especiales del sistema.

Los diferentes **tipos de usuarios** en Linux:

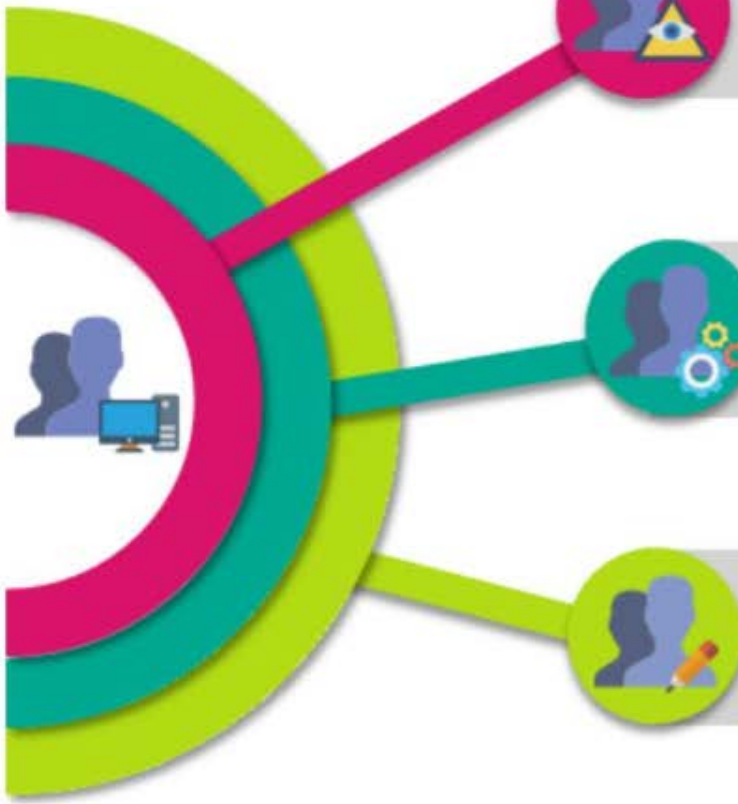
- **Root o superusuario** (#) (***`UID=0`***): es el único que tiene privilegios sobre todo el sistema, y responsable de las tareas de administración del sistema, tales como la instalación y desinstalación de software, entre otras muchas. Para cualquier acción que necesite permisos de superusuario, el sistema requerirá credenciales root.
- **Usuarios especiales de sistema**: van vinculados a ciertos servicios, y que pueden asumir ciertos privilegios relativos a este servicio. Se crean automáticamente en la instalación del sistema o de ciertas aplicaciones. Por ejemplo *mail, apache, syslog..*
- **Usuarios estándar** ($)**:** Las cuentas de usuarios individuales con login. Puede haber tantas como se requiera. Cada usuario estándar posee su directorio personal dentro de `/home`

Usuarios y grupos locales

Cada usuario tiene asociada una **cuenta de usuario** concreta (en este curso solo veremos las cuentas locales).

Dichas cuentas no solo ofrecen al usuario un nombre y una contraseña, también le proporciona una ruta para almacenar sus documentos y su perfil, generalmente dentro de la carpeta **/home/nombre-usuario** y comúnmente denominada carpeta home del usuario, y un intérprete de comandos (***shell***) que le permitirá ejecutar aplicaciones.

Usuarios y grupos locales



ADMINISTRADOR, ROOT O SUPERUSUARIO (UID=0)

Es el único que tiene privilegios sobre todo el sistema, y el responsable de las tareas de administración del sistema.

USUARIO DEL SISTEMA

Vinculados a ciertos servicios y con ciertos privilegios sobre estos. Se crean automáticamente durante instalación del sistema o de ciertas aplicaciones: *bin*, *mail*, *apache*, *clamav*, *syslog*, etc.

USUARIO ESTÁNDAR

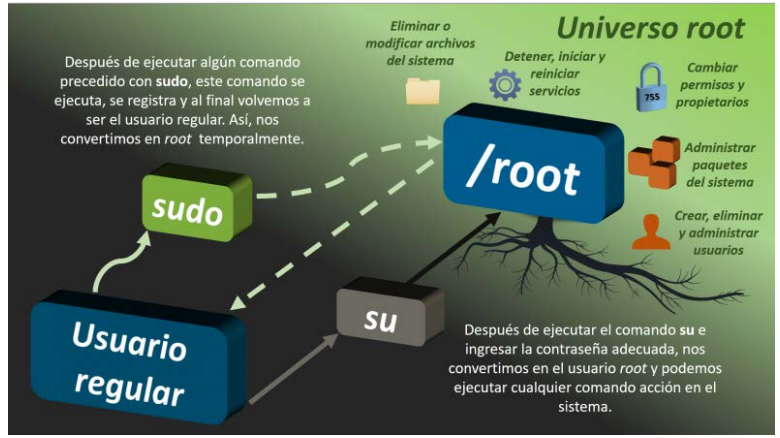
Cada uno posee su directorio personal dentro de */home*, donde se almacenan los archivos personales, además de preferencias para algunas aplicaciones e incluso archivos temporales.

Usuarios y grupos locales

Superusuario root

Como hemos comentado el usuario **root** (UID=0) en Linux es el usuario que posee mayor nivel de privilegios en el sistema. Es el único que tiene privilegios sobre todo el sistema, y el responsable de las tareas administrativas: instalaciones, modificaciones, configuraciones, etc.

Mediante el comando **sudo** en la línea de comandos se permite al usuario actual, ejecutar aplicaciones o procesos bajo los privilegios de root u otro usuario.



Iniciar sesión de continuo como usuario root es una práctica poco recomendada (`sudo su`) ya que mantener abierta permanentemente una sesión de usuario root podría traer graves consecuencias e inutilizar el sistema.

Usuarios y grupos locales

Ficheros importantes

La información sobre usuarios, grupos y contraseñas se guarda en los archivos:

- `/etc/passwd`: información sobre usuarios
- `/etc/group`: información sobre grupos
- `/etc/shadow`: contraseñas cifradas

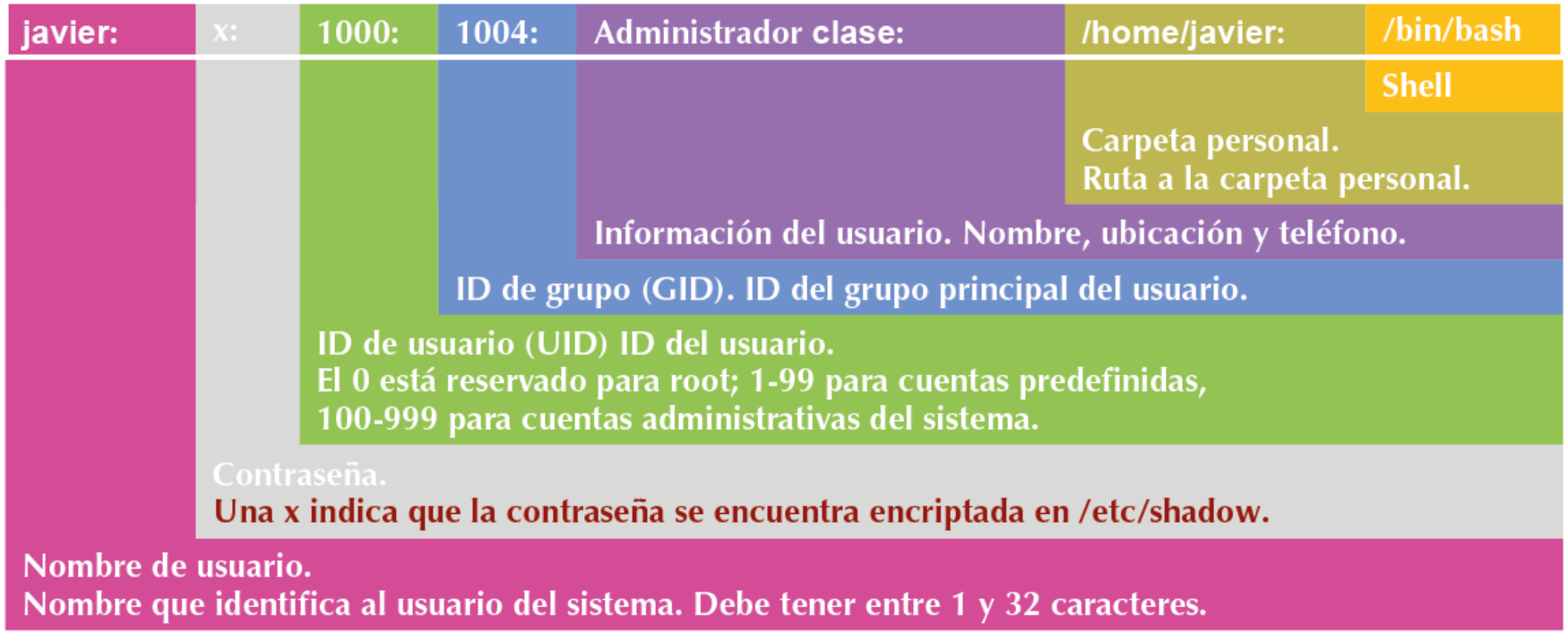
Dichos ficheros contiene diversos campos separados por dos puntos que estudiaremos a continuación y se puede visualizar su contenido mediante **cat**.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
```

Usuarios y grupos locales

Ficheros importantes

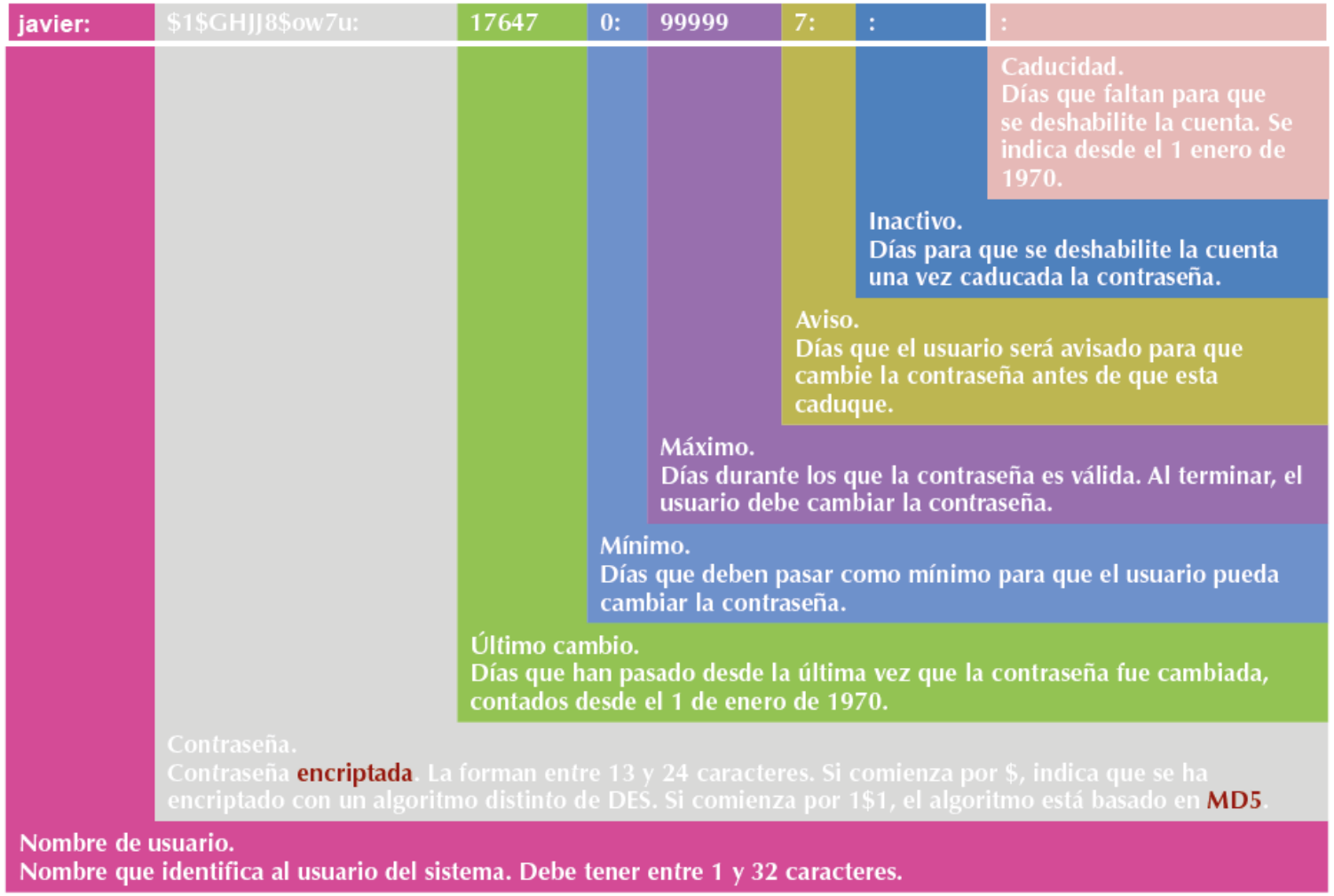
Estructura del fichero **`/etc/passwd`** con el significado de cada para uno de sus campos separados por dos puntos:



Usuarios y grupos locales

Ficheros importantes

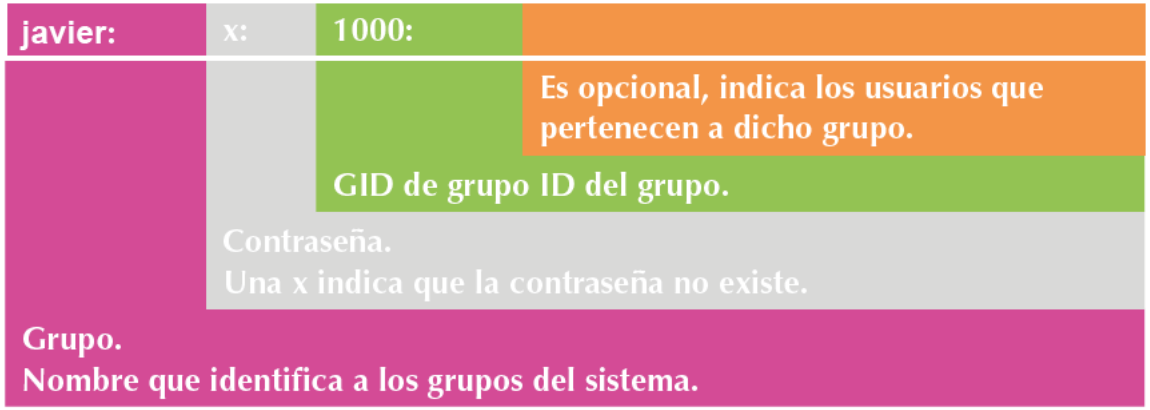
Estructura del fichero **`/etc/shadow`** con el significado de sus campos:



Usuarios y grupos locales

Ficheros importantes

Estructura del fichero **`/etc/group`** con el significado de sus campos:



Para mostrar la información de duración de contraseñas de forma más compresible que en el fichero **/etc/shadow** se puede usar el comando **chage –l usuario**

```
javi@javi-VirtualBox:~$ chage -l javi
Último cambio de contraseña                                   : abr 28, 2019
La contraseña caduca                                  : nunca
Contraseña inactiva                                   : nunca
La cuenta caduca                                              : nunca
Número de días mínimo entre cambio de contraseña              : 0
Número de días máximo entre cambio de contraseña              : 99999
Número de días de aviso antes de que caduque la contraseña    : 7
```

Usuarios y grupos locales

Privilegios de administrador

Es posible que un usuario estándar necesite disponer de **privilegios de administrador** y poder realizar acciones como **sudo**.

Ello se puede llevar a cabo de dos formas:

1. Editando el fichero **`/etc/sudoers`**.

   Añadir el usuario al fichero, por lo que habría que añadir una línea como esta:

   ```
   Nombre_usuario ALL=(ALL:ALL) ALL
   ```

2. Editando el fichero **`/etc/group`**

   Si, en el fichero `/etc/sudoers` ya existe una línea referente al grupo *sudo* (o *admin)* como se debe incluir al usuario a uno u otro grupo como secundario editando el fichero `/etc/group` (o agregarlo mediante `usermod`) a dichos grupos.

Usuarios y grupos locales

Comandos de usuarios

Los comandos más importantes para la gestión de **usuarios locales**:

Comando	Acción	Ejemplo
useradd	agregar usuarios	`useradd -g users usuario`
adduser	agregar usuario mediante un script*	`adduser usuario`
usermod	modificar usuarios	`usermod -d /home/javi user`
userdel	borrar usuarios	`userdel -r usaurio`
deluser	borrar usuario mediante un script*	`deluser usuario`
passwd	establecer contraseña a un usuario	`passwd usuario`
chage	visualizar o establecer duración de contraseñas	`chage -l usuario`
who	usuarios actualmente logueados.	`who o w`
id	Muestra la identidad del usuario (UID) y sus grupos	`id usuario`

`adduser` y **`deluser`** utilizan un script más elaborado para agregar o eliminar usuarios

Usuarios y grupos locales

Agregar usuarios/grupos

Para **agregar** un usuario de forma sencilla usaremos **useradd.** Su sintaxis:

```
useradd [parámetros] [nombre_de_usuario]
```

Parámetros:

```
-d   Especifica el directorio home del usuario
-s   Especifica el shell del usuario
-g   Especifica el grupo primario del usuario
-G   Especifica los grupos secundarios del usuario
-uid Especifica el identificador de usuario para el usuario
```

Después de crear un usuario con useradd conviene siempre recordar que hay que asignarle a continuación contraseña con el comando **passwd.**

Para **agregar** grupos se utiliza **groupadd:**

```
groupadd [opciones] [nombre_de_usuario]
```

Parámetros:

```
-g GID Especifica ID para el grupo, el cual debe ser único y mayor que 499
```

Usuarios y grupos locales

Modificar usuarios

Para **modificar** las propiedades de un usuario ya existente se utiliza **usermod** :

```
usermod [parámetros] [nombre_de_usuario]
```

Parámetros:

```
-C    Agrega un nuevo valor al campo de comentarios del usuario
-d    Especifica un nuevo directorio home para el usuario
-g    Cambiar el grupo primario del usuario
-G    Cambiar los grupos secundarios del usuario
-s    Cambiar el Shell por defecto del usuario
-L    Bloquea la contraseña del usuario
-u    Cambiar el UID asignado (siempre > 1000)
-e    Fecha en la que estará deshabilitada la cuenta de usuario (formato AAAA-MM-DD)
```

Usuarios y grupos locales

Gestión de contraseñas

Para la **gestión de contraseñas** se utiliza el comando **chage** con la siguiente sintaxis:

```
chage [parámetros] [nombre_de_usuario]

Parámetros:
-l Muestra información sobre la antigüedad de la cuenta
-d Establece el último día que se cambió la contraseña. Si se pone valor 0
   obligará al usuario a cambiar la contraseña en su próximo acceso.
-m Establece el número mínimo de días entre cambios de contraseña. Si se
   establece en un valor de 0, indica que el usuario puede cambiar su contraseña
   en cualquier momento.
-M Establece el número máximo de días durante los cuales una contraseña es
   válida.
-E Establece una fecha concreta de caducidad de una cuenta escrita en formato
   AAAA-MM-DD
```

Usuarios y grupos locales

Gestión de contraseñas

Para la **modificación de contraseñas** se utiliza el comando **`passwd`** con la sintaxis:

```
passwd [parámetros] [nombre_de_usuario]
```

Parámetros:

```
-a informa del estado de las contraseñas de todas las cuentas
-d borra la contraseña para la cuenta indicada
-l bloquea la contraseña de la cuenta indicada
-u desbloquea la contraseña de la cuenta indicada
-w establece el aviso de caducidad a los días indicados
-n establece el número mínimo de días antes de que se cambie la contraseña
```

Usuarios y grupos locales

Uso de los comandos

useradd es un comando que ejecuta un binario del sistema, mientras que **adduser** es un script en *perl* que utiliza el comando useradd. Es más completo ya que crea el *home, pide contraseña* y hace verificaciones pero no está en todas las distribuciones. Igual con **deluser** que además elimina sus ficheros y directorios del home.

Crear al usuario con useradd es el primer paso, el segundo es asignarle una contraseña. Esto se logra con el comando **passwd**, que permite ingresar la contraseña y su verificación. Para verificar el *id* del usuario y grupo usar **id**.

```
javi@javi-VirtualBox:~$ sudo adduser norah
Añadiendo el usuario `norah' ...
Añadiendo el nuevo grupo `norah' (1001) ...
Añadiendo el nuevo usuario `norah' (1001) con grupo `norah' ...
Creando el directorio personal `/home/norah' ...
Copiando los ficheros desde `/etc/skel' ...
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para norah
Introduzca el nuevo valor, o presione INTRO para el predeterminado
        Nombre completo []:
        Número de habitación []:
        Teléfono del trabajo []:
        Teléfono de casa []:
        Otro []:
¿Es correcta la información? [S/n]
```

```
javi@javi-VirtualBox:~$ sudo passwd norah
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
javi@javi-VirtualBox:~$
```

```
javi@javi-VirtualBox:~$ id norah
uid=1001(norah) gid=1001(norah) grupos=1001(norah)
javi@javi-VirtualBox:~$
```

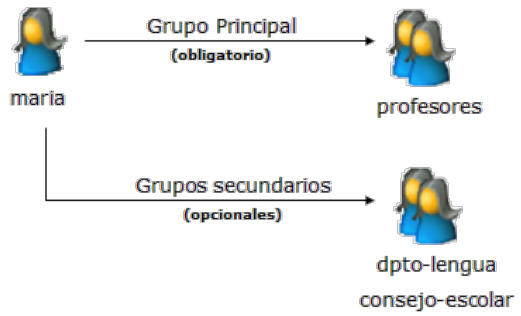
passwd tiene varias opciones que permiten bloquear la cuenta `-l`, desbloquearla `-u`, y varias opciones más que controlan la vigencia de la contraseña, es decir, es otro modo de establecer los valores de la cuenta en `/etc/shadow`.

Usuarios y grupos locales

Comandos de grupos

Los comandos más importantes para la gestión de **grupos**:

Comando	Acción	Ejemplo
groupadd	crear un grupo	`groupadd alumnos`
groupmod	modificar grupo	`groupmod -g 2000 profesores`
groupdel	borrar grupo	`groupdel profesores`
adduser	añadir usuario a un grupo	`adduser juan profesores`
deluser	eliminar usuario de un grupo	`deluser juan profesores`
groups	ver los grupos a los que pertenece un usuario	`groups juan`

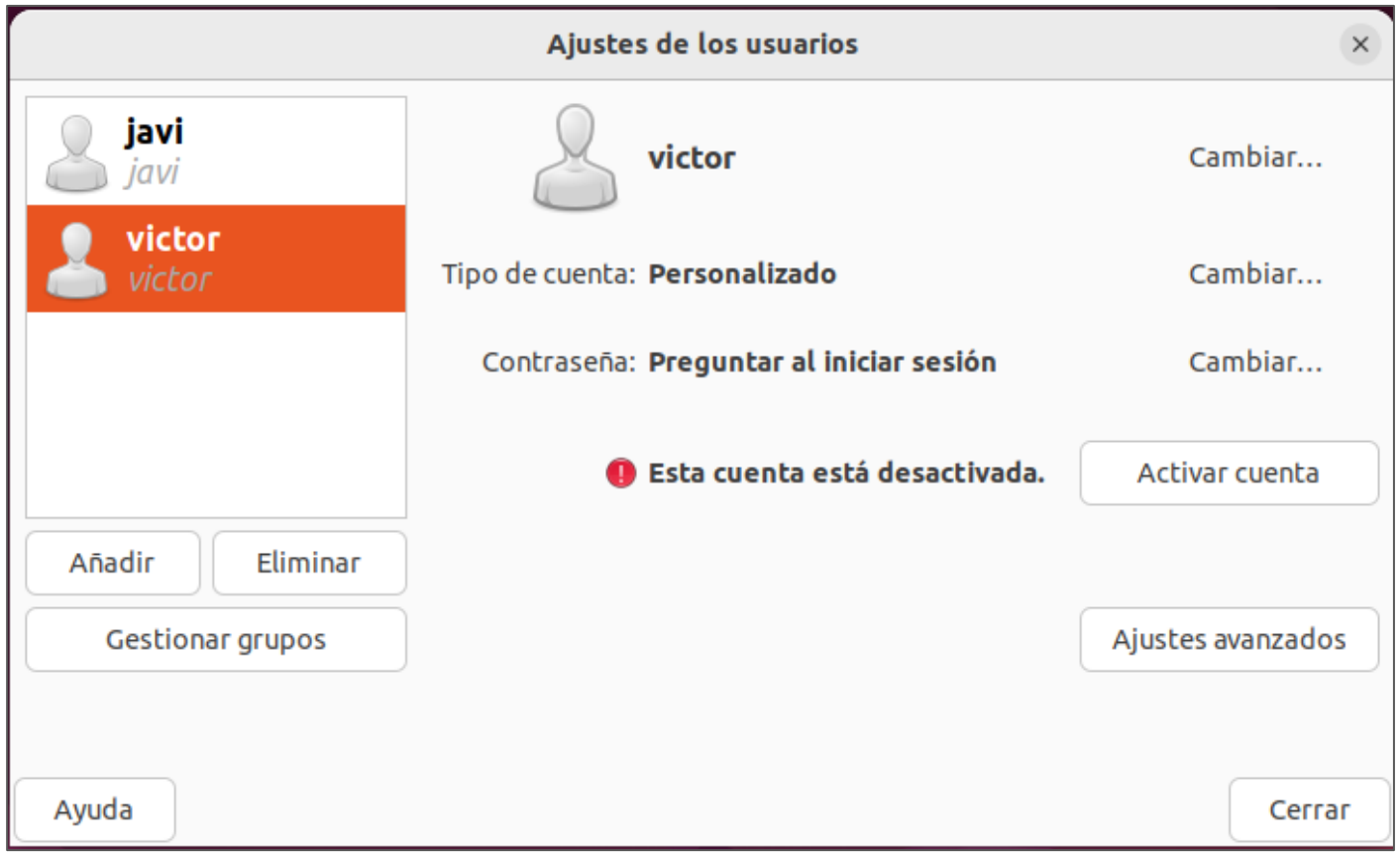


Todos los usuarios pertenecen al menos a un grupo que es el **grupo principal del usuario**, también llamado grupo primario del usuario, pero pueden pertenecer a más grupos. En caso de que pertenezcan a más grupos, éstos serán **grupos secundarios**.

El fichero de contraseñas de usuarios es el `/etc/gshadow`

Usuarios y grupos locales

Ubuntu dispone a su vez de una herramienta gráfica de administración de usuarios que es **users-admin** (paquete *`system-tools`*).
Para ejecutarla podemos abrir una consola de root y ejecutar `users-admin` o desde el menú *configuración>cuentas de usuario*



Permisos

Los **permisos** en Linux funcionan utilizando el mismo esquema que en Unix; se aplican sobre archivos o sobre directorios y van asociados a usuarios o grupos pudiendo ser de lectura, de escritura o de ejecución.

Todos los archivos y directorios en Linux tienen asociado un conjunto de permisos que debe definir las posibilidades de lectura, escritura y ejecución que se aplican al usuario propietario del archivo, al grupo de usuarios al que pertenece, y al resto del mundo.

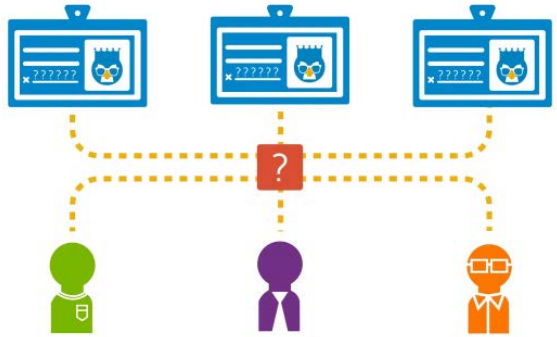
Existen por tanto tres tipos de permisos **generales** en Linux ya conocidos:

- Permiso de lectura **(r)**
- Permiso de escritura **(w)**
- Permiso de ejecución **(x)**

Permisos

En la siguiente tabla se muestran los permisos necesarios para ejecutar determinados comandos conocidos:

Comando	Permisos directorio origen	Permisos fichero	Permisos directorio destino
`cd`	x	No aplicable	No aplicable
`ls`	r	No aplicable	No aplicable
`mkdir`	w, x	No aplicable	No aplicable
`rmdir`	w, x	No aplicable	No aplicable
`cat`	x	r	No aplicable
`rm`	w, x		No aplicable
`cp`	x	r	w, x
`mv`	w, x		w, x



Permisos

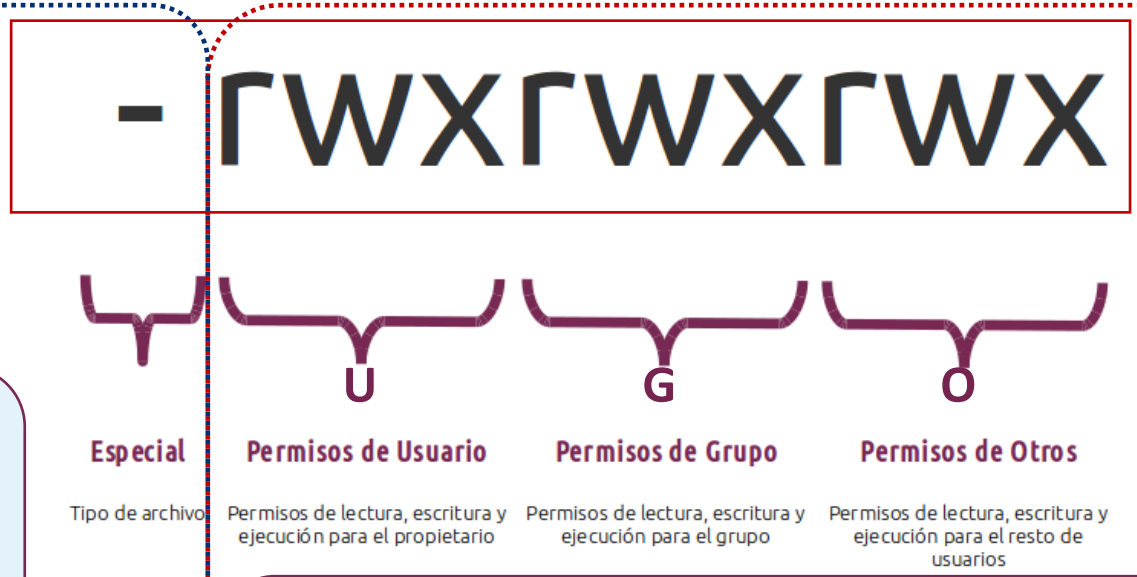
Privilegios

De entrada, en cualquier fichero o directorio en Linux hay tres tipos o niveles de **privilegios** a los que van dirigidos los permisos anteriores:

- **Permisos del Usuario (U).** Es el primer nivel de privilegios. Básicamente representa los permisos que se aplican al propietario de un determinado fichero o directorio.
- **Permisos del Grupo (G)**. Estamos ante el segundo nivel de privilegios, que define los derechos de lectura, escritura y ejecución que se aplican solo a aquellos usuarios que pertenecen al mismo grupo que el propietario del fichero.
- **Permisos de Otros (O)**. Este es el último nivel. Representan los privilegios de lectura, escritura y ejecución por parte del resto de usuarios que no entran en ninguno de los niveles anteriores.

Permisos

Representación de permisos



-	archivo
d	directorio
l	enlace
b	disp. bloque
s	socket
p	tubería

r	Tiene permisos de lectura
w	Tiene permisos de escritura
x	Tiene permisos de ejecución
-	No tiene permiso correspondiente

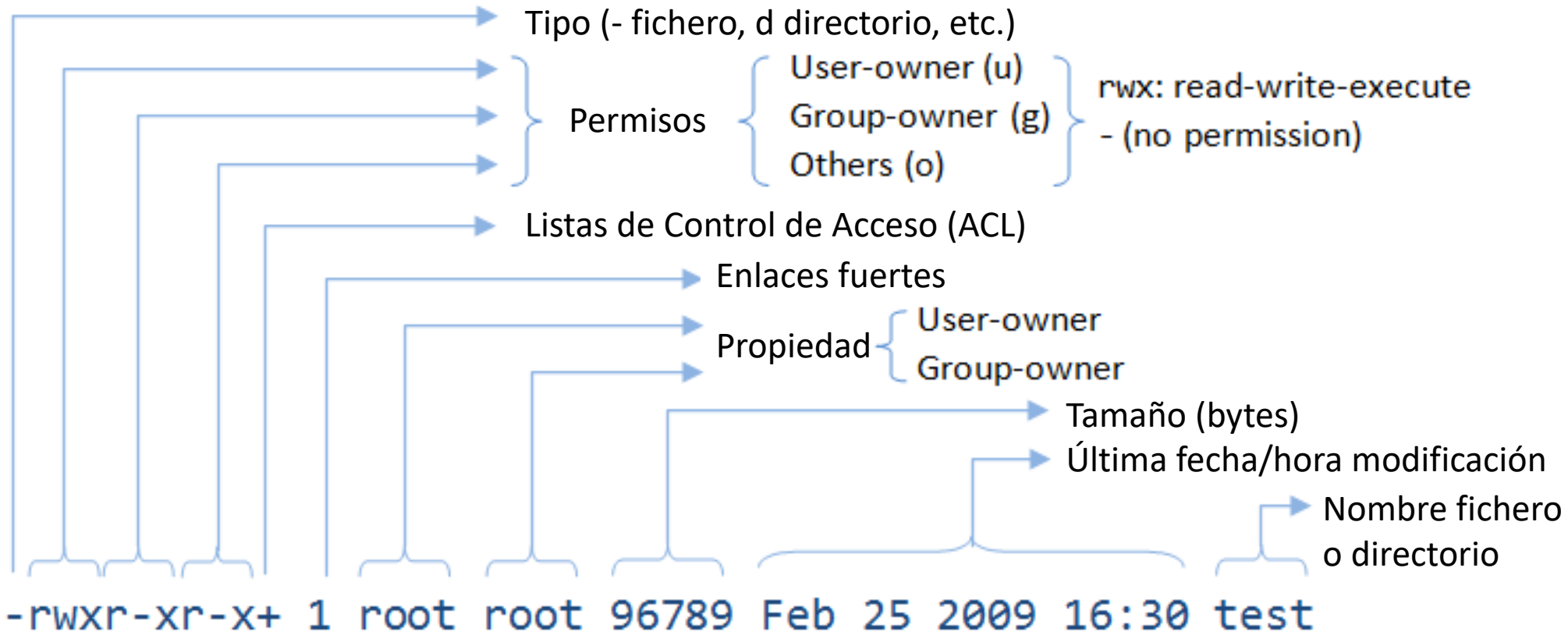
−|d..	r	w	x	r	w	x	r	w	x
Tipo de fichero	Permisos para el dueño del fichero			Permisos para el grupo al que pertenece el fichero			Permisos para el resto de usuarios		

Permisos

Representación de permisos

Cada archivo en Linux queda identificado por 10+1 caracteres mismos a los que se les denomina máscara. Se pueden mostrar mediante el comando `ls -l`

El significado de todos los **campos y permisos** en dicho comando será:



Permisos

Representación de permisos

El significado de los **permisos** varía en caso de tratarse de un archivo o de un directorio, lo cual puede ser problemático:

Permiso	Aplicación	Descripción
Lectura (R)	Archivo	Lectura del contenido del archivo.
Lectura (R)	Directorio	Listar los archivos que están contenidos en el directorio (ls).
Escritura (W)	Archivo	Lectura y modificación del contenido del archivo. Eliminación del archivo.
Escritura (W)	Directorio	Eliminar carpeta. Crear subdirectorios dentro de ella.
Ejecución (X)	Archivo	Ejecución de archivos binarios
Ejecución (X)	Directorio	Posicionarnos (cd). Acceso a propiedades y contenido de archivos y subdirectorios.

Permisos

Representación numérica permisos

También existe una representación numérica en **octal**, que parte de idea de la representación anterior, pero sustituye cada grupo ***r w x*** (y los especiales) por un valor **numérico que sumar** en caso de tener dicho permiso.

Permisos

La siguiente tabla muestra la forma de asignar permisos en ambas notaciones:

Permisos normales		Valor octal	Notación simbólica
Propietario:	Lectura	400	u+r
	Escritura	200	u+w
	Ejecución / Acceso	100	u+x
Grupo:	Lectura	40	g+r
	Escritura	20	g+w
	Ejecución / Acceso	10	g+x
Otros:	Lectura	4	o+r
	Escritura	2	o+w
	Ejecución / Acceso	1	o+x

Permisos

Modificación de permisos

Habiendo entendido lo anterior, es ahora fácil cambiar los permisos de cualquier archivo o directorio, utilizando para ello el comando **`chmod`** (*change mode*), cuya sintaxis es la siguiente:

```
chmod [parámetros] permisos archivo
```

El parámetro **–R** aplicará los permisos de forma recursiva a los subdirectorios.

- Existen dos variantes para aplicar permisos, usando letras o los valores octales como veremos con ejemplos a continuación:

  - 1ª forma: `chmod [u,g,o][+,-][r,w,x] ruta-directorio`
  - 2ª forma: `chmod [valor] ruta-directorio`

Permisos

Modificación de permisos

- **1ª Forma**: `chmod [u,g,o][+,-][r,w,x] ruta-directorio`

Dar permiso de escritura al usuario propietario sobre `'examen.txt'`:

Quitar permiso de escritura al resto de usuarios sobre el archivo:

Dar permiso de ejecución al grupo propietario sobre todos los archivos:

Dar permiso de lectura al grupo propietario sobre `'examen.txt'`:

Dar permiso de escritura al usuario y quitar el de lectura a grupos y otros:

Dar permisos de escritura a usuarios y grupos:

Permisos

Modificación de permisos

- **2ª Forma**: `chmod [valor] ruta-directorio`

Dar todos los permisos al usuario y ninguno al grupo ni al resto del fichero:

Dar a usuario y grupo permisos de lectura y ejecución, ninguno al resto:

Dar todos los permisos al usuario así como lectura y ejecución a los demás:

Dar todos los permisos al usuario y de lectura al resto, sobre todos los archivos de texto que contenga el directorio:

Dar permisos completos a todos los usuarios al directorio /respaldo/

Cambiar permisos a lectura y escritura a todos los archivos incluyendo subdirectorios:

Permisos

Máscaras

Es importante saber que al crear un fichero o directorio, se creará por unos permisos por defecto llamados **máscara** que pueden visualizarse o modificarse a través del comando **`umask`.**

Normalmente el valor por defecto de **umask** suele ser **002**. Podemos averiguarlo simplemente ejecutando el comando umask.

Los ficheros y directorios tienen definidos por defecto unos permisos base, **666** para los ficheros y **777** para los directorios.

Este será por tanto el valor a restar usando el resultado de la **máscara** devuelto por `umask`:

- Por eso, al crear un fichero, su permiso inicial será **664** (`-rw-r--r--`), valor resultante de realizar la operación *666 - **002***.
- Al crear un directorio, su permiso inicial será **775** (`drwxr-xr-x`), resultado de *777 - **002***.

Permisos

Máscaras

```
[gacanepa@server ~]$ umask
0002
[gacanepa@server ~]$ touch file1
[gacanepa@server ~]$ mkdir dir1
[gacanepa@server ~]$ ls -ld file1 dir1
drwxrwxr-x 2 gacanepa gacanepa 4096 Aug  4 23:34 dir1
-rw-rw-r-- 1 gacanepa gacanepa    0 Aug  4 23:34 file1
[gacanepa@server ~]$ rm file1
rm: remove regular empty file ‘file1’? y
[gacanepa@server ~]$ rmdir dir1
[gacanepa@server ~]$ umask 000
[gacanepa@server ~]$ touch file1
[gacanepa@server ~]$ mkdir dir1
[gacanepa@server ~]$ ls -ld file1 dir1
drwxrwxrwx 2 gacanepa gacanepa 4096 Aug  4 23:34 dir1
-rw-rw-rw- 1 gacanepa gacanepa    0 Aug  4 23:34 file1
[gacanepa@server ~]$
```

777-002
666-002

777-000
666-000

Permisos

Modificación de propietario

El comando **chown** (*change user*) permite cambiar el propietario de un archivo o directorio en sistemas tipo UNIX. Puede especificarse tanto el nombre de un usuario, así como el identificador de usuario (UID) y el identificador de grupo (GID).

Opcionalmente, utilizando un signo de dos puntos (:), o bien un punto (.), sin espacios entre ellos, entonces se cambia el **usuario y grupo** al que pertenece cada archivo. Con el parámetro –R se hará el cambio de propietario de forma recursiva.

```
chown [parámetros] archivo
```

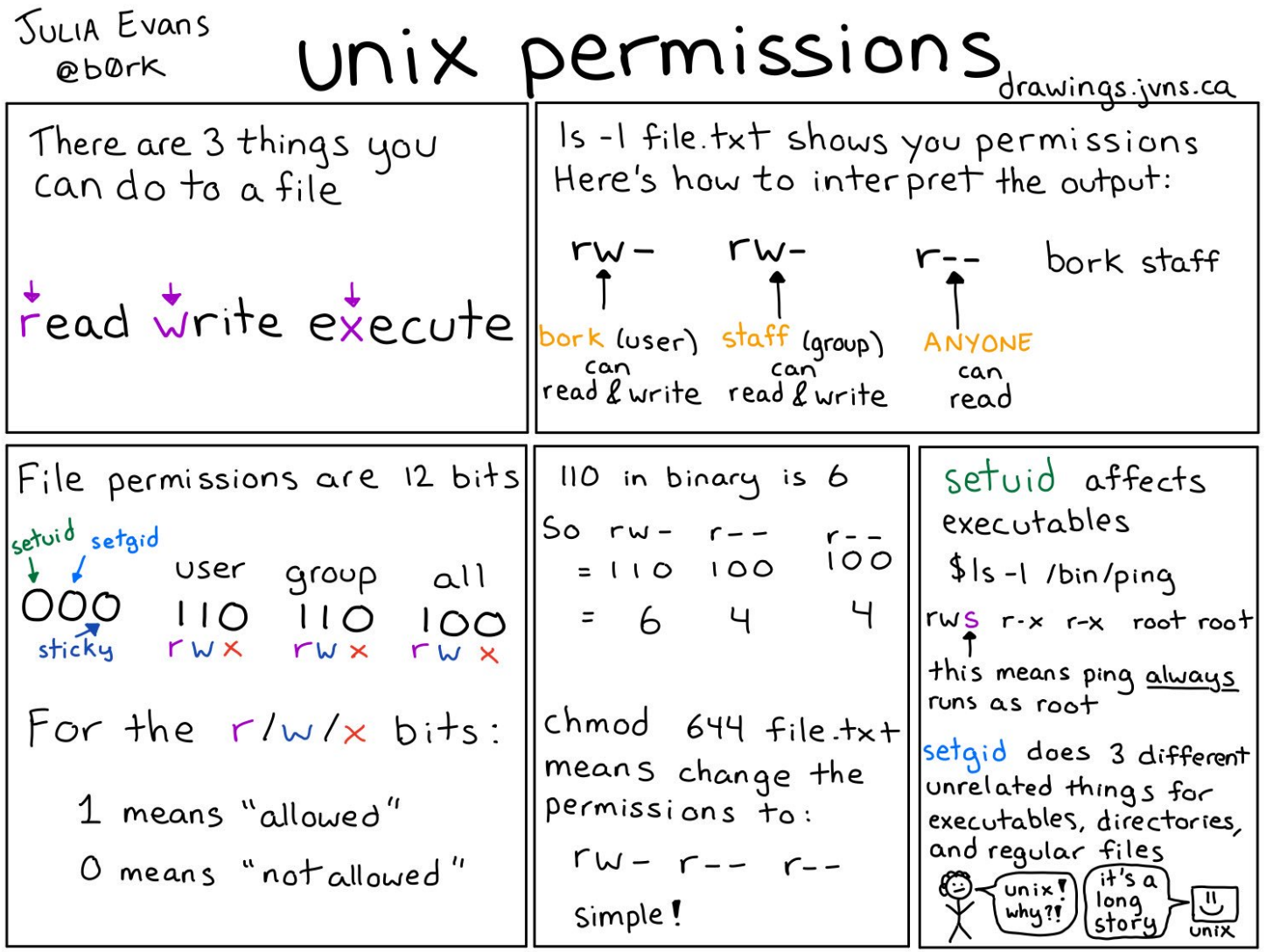
```
# cambiar usuario propietario
chown root tmpfile
# cambiar grupo propietario
chown :profesores tmpfile
# cambiar usuario y grupo propietario
chown root:profesores tmpfile
```

Permisos

Comandos importantes para la gestión de **permisos**:

Comando	Acción	Ejemplo
chmod	cambiar permisos fichero o directorio	`chmod 750 mifichero.txt`
chown	cambiar propietario fichero o directorio	`chown root prueba.txt`
chgrp	cambiar grupo fichero o directorio	`chgrp alumnos notas.doc`
umask	máscara de permisos por defecto	`umask 025`

Permisos

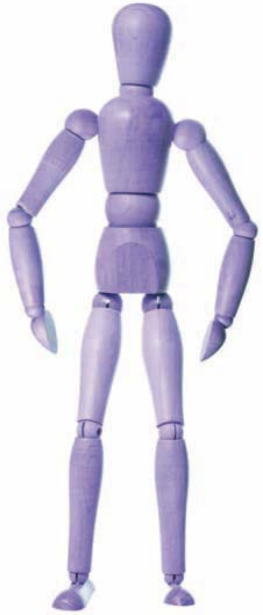

Julia Evans
@b0rk
unix permissions
drawings.jvns.ca
There are 3 things you can do to a file
read write execute
ls -l file.txt shows you permissions
Here's how to interpret the output:
rw- rw- r-- bork staff
bork (user) can read & write
staff (group) can read & write
ANYONE can read
File permissions are 12 bits
setuid setgid
000 user 110 group 110 all 100
sticky
rwx rwx rwx
For the r/w/x bits:
1 means "allowed"
0 means "not allowed"
110 in binary is 6
So rw- r-- r--
= 110 100 100
= 6 4 4
chmod 644 file.txt means change the permissions to:
rw- r-- r--
Simple!
setuid affects executables
$ls -l /bin/ping
rws r-x r-x root root
this means ping always runs as root
setgid does 3 different unrelated things for executables, directories, and regular files
unix! why?!
it's a long story
unix

Creación de plantillas de usuarios

Las **plantillas de usuarios** permiten ahorrar tiempo en la creación de usuarios cuando su número aumenta. Son como moldes que poseen la configuración básica para crear a partir de ellos el resto de usuarios.

La creación de plantillas de usuarios en Linux se basa en la utilización de:

- **Archivos** que se copian en la creación de un usuario para configurar su entorno personal. Estos archivos se configuran en el directorio `/etc/skel/`
- **Parámetros** que se dan por defecto a los usuarios en su creación. Es decir, nosotros podemos especificar en la creación los parámetros de los usuarios, pero si no lo hiciéramos estos serían tomados del archivo `/etc/login.defs` visto anteriormente.



Creación de plantillas de usuarios

Cuando se crea un **usuario**, el sistema realiza lo siguiente:

1. Copia los archivos que se encuentran en el directorio `/etc/skel` en el directorio de inicio del nuevo usuario.
2. Establece al usuario y al grupo del usuario que se está creando como propietario y grupo de estos archivos copiados.

Gracias a esto, cuando creamos los usuarios podemos controlar la apariencia de su escritorio, sus variables de entorno, sus alias y muchos otros parámetros de su entorno.