

# **UNIDAD 9: LINUX. GESTIÓN DE USUARIOS Y GRUPOS. GESTIÓN DE PROCESOS**

**Sistemas Operativos Monopuesto  
Curso 2012/2013**

# UT9: LINUX. GESTIÓN DE USUARIOS Y GRUPOS. GESTIÓN DE PROCESOS

- 1. Usuarios y Grupos
- 2. Permisos
- 3. Concepto de Procesos
- 4. Gestión de Procesos
- 5. Ejercicios

# 1. USUARIOS Y GRUPOS

## ○ UID

- Número entero distinto para cada usuario, que lo identifica específicamente en el sistema

## ○ Grupo:

- Conjunto de usuarios identificados con su GID (distinto y específico para cada grupo)
- Cada usuario pertenece a un grupo principal (obligatorio)
- Un usuario puede pertenecer a más grupos secundarios

# 1. USUARIOS Y GRUPOS

## ○ Tipos de usuarios

- Usuario root
- Usuarios del sistema
- Usuarios normales

# 1. USUARIOS Y GRUPOS. FICHEROS ESPECIALES

## ○ Fichero *passwd*

- Todos los usuarios pueden acceder a él pero sólo el administrador puede modificarlo
- Se encuentra en el directorio */etc* y en cada línea contiene la información de los usuarios dados de alta en el sistema, con campos separados por (:)
  - Primer campo. Contiene el nombre de presentación (de uno a siete caracteres para los usuarios)
  - Segundo campo. Contiene una x. Indica que el usuario tiene una contraseña cifrada. Esta contraseña se encontrará en el fichero */etc/shadow*.
  - Tercer y cuarto campo. Hacen referencia al ID de usuario y del Grupo respectivamente

# 1. USUARIOS Y GRUPOS. FICHEROS ESPECIALES

## ○ Fichero *passwd*

- Quinto campo. Para poner comentarios
- Sexto campo. Contiene el nombre del directorio en el que se situará el usuario tras la conexión al sistema. Es el valor inicial de la variable *HOME*
- Séptimo campo. Indica al sistema el programa que tiene que ejecutar de forma automática. Se denomina *shell de presentación del usuario*
- La primera línea ya que hace referencia al usuario root o súper usuario. El ID de root es siempre 0. Su directorio de trabajo es el directorio raíz (/root) y el *shell* que carga por defecto es *bash*

# 1. USUARIOS Y GRUPOS. FICHEROS ESPECIALES

## ○ Fichero *shadow*

- Este fichero es uno de los ficheros más importantes de configuración y de identificación de usuarios ante el sistema
- Solo un usuario root o con permisos de administración puede leerlo
- Cada línea tiene una serie de campos separados por el carácter ( : )
- Contiene información acerca de la contraseña de cada usuario, datos referentes a la caducidad y envejecimiento de la misma.
- Este fichero se encuentra en el directorio `/etc` y contiene una línea por cada línea del archivo `/etc/passwd`

# 1. USUARIOS Y GRUPOS. FICHEROS ESPECIALES

## ○ Fichero *shadow*

- Analizado por campos, cada uno de ellos significaría lo siguiente:
  - **Primer campo.** Nombre de presentación
  - **Segundo campo.** Este campo puede contener lo siguiente:
    - Contraseña cifrada para ese nombre de presentación. Su estructura cambia en función del algoritmo de encriptación usado
    - \* y ! indica que no hay contraseña. No se deja en blanco para evitar que cualquier usuario pueda escribir un valor de contraseña.



# 1. USUARIOS Y GRUPOS. FICHEROS ESPECIALES

## ○ Fichero *shadow*

- Analizado por campos, cada uno de ellos significaría lo siguiente:
  - **Tercer campo.** Número de días transcurrido que entre el 1 de enero de 1970 y el día en que la contraseña fue modificada por última vez
  - **Cuarto campo.** Número mínimo de días requeridos para que la contraseña pueda ser cambiada por el usuario
  - **Quinto campo.** Número máximo de días que una contraseña es válida. Pasado ese periodo el usuario es forzado a cambiarla
  - **Sexto campo.** Número de días de antelación con el que el sistema avisa al usuario que su contraseña va a espirar

# 1. USUARIOS Y GRUPOS. FICHEROS ESPECIALES

## ○ Fichero *shadow*

- Analizado por campos, cada uno de ellos significaría lo siguiente:
  - **Séptimo campo.** Número de días de inactividad que el usuario le permite al usuario. Si el usuario no se conecta al sistema en ese intervalo de días, el sistema no le permitirá la conexión
  - **Octavo campo.** Días (en fecha absoluta) desde el 1 de enero de 1970 a partir de cual el nombre de presentación no podrá ser utilizado
  - **Noveno campo.** No utilizado, reservado para usos futuros

# 1. USUARIOS Y GRUPOS. FICHEROS ESPECIALES

## ○ Fichero group.

- Contiene información sobre los grupos de usuarios de nuestro sistema, y que usuarios pertenecen a cada grupo
- El grupo principal de cada usuario es aquel cuyo GID viene en el fichero /etc/passwd
- Está en el directorio **/etc**, al igual que los anteriores
- Tiene los siguientes campos:
  - **Primer campo.** Nombre del grupo
  - **Segundo campo.** No usado actualmente (antes almacenaba la contraseña del grupo: /etc/gshadow)
  - **Tercer campo.** Número identificativo del grupo (GID)
  - **Cuarto campo.** Lista de usuarios pertenecientes al grupo

# 1. USUARIOS Y GRUPOS. FICHEROS ESPECIALES

## ○ **/etc/gshadow**

- Contiene las contraseñas de los grupos del sistema:
  - Encriptadas
  - \*
  - !

## ○ **/etc/default/useradd**

- Valores por defecto usados cuando se crea un usuario nuevo con el comando useradd

## ○ **/etc/useradd.conf**

- Valores por defecto usados cuando se crea un usuario nuevo con el comando adduser

# 1. USUARIOS Y GRUPOS. FICHEROS ESPECIALES

## ○ **/etc/deluser.conf**

- Valores por defecto usados cuando se elimina un usuario con el comando deluser

## ○ **/etc/login.defs**

- Define algunos valores por defecto sobre la encriptación de contraseñas y otros parámetros necesarios al generar nuevos usuarios.

## ○ **/etc/shells**

- Lista de shells válidos en el sistema

# 1. USUARIOS Y GRUPOS. ENTORNO GRÁFICO

- Sistema – Administración – Usuarios y Grupos
  - Usuarios



# 1. USUARIOS Y GRUPOS. ENTORNO GRÁFICO

## ○ Sistema – Administración – Usuarios y Grupos

- El sistema pide que se vuelva a introducir la contraseña cada vez que realicemos una acción de administración:



# 1. USUARIOS Y GRUPOS. ENTORNO GRÁFICO

## ○ Sistema – Administración – Usuarios y Grupos

- Podemos añadir nuevos usuarios, eliminarlos, etc. utilizando los distintos botones que aparecen en el menú





# 1. USUARIOS Y GRUPOS. ENTORNO GRÁFICO

- Creación de un nuevo usuario:

Cuenta de usuario nueva

Cuenta Contacto Privilegios del usuario Avanzado

**Configuración básica**

Usuario: usuario2

Nombre real: usuario2

Perfil:

- Administrador
- Usuario del escritorio
- Usuario sin privilegios

**Contraseña**

☐ Establecer contraseña

Contraseña del usuario:

Confirmación:

☒ Generar contraseña aleatoria

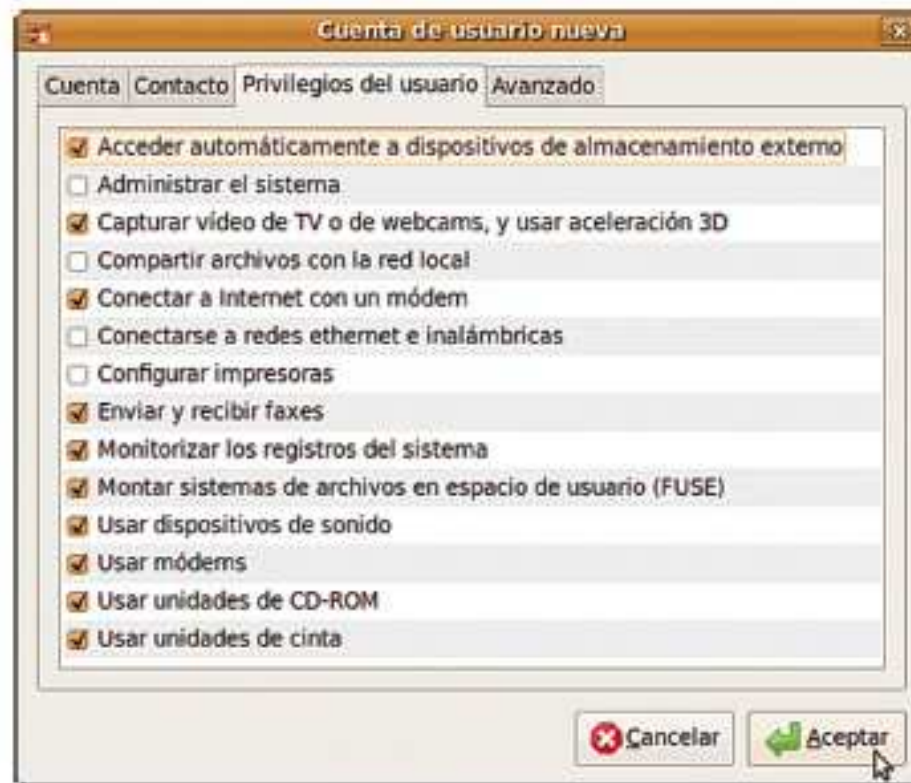
Contraseña establecida a: YTmR50xY

Generar

Cancelar Aceptar

# 1. USUARIOS Y GRUPOS. ENTORNO GRÁFICO

- Creación de un nuevo usuario:



# 1. USUARIOS Y GRUPOS. ENTORNO GRÁFICO

- Creación de un nuevo usuario:



# 1. USUARIOS Y GRUPOS. ENTORNO GRÁFICO

- Sistema – Administración – Usuarios y Grupos
  - Grupos



# 1. USUARIOS Y GRUPOS. MODO TEXTO

- **su.** Cambia de usuario. Su sintaxis es:

**su [usuario]**

- Donde:

- usuario es el login del usuario con el que nos queremos validar
- Si no se pone nada se asume que el usuario es root

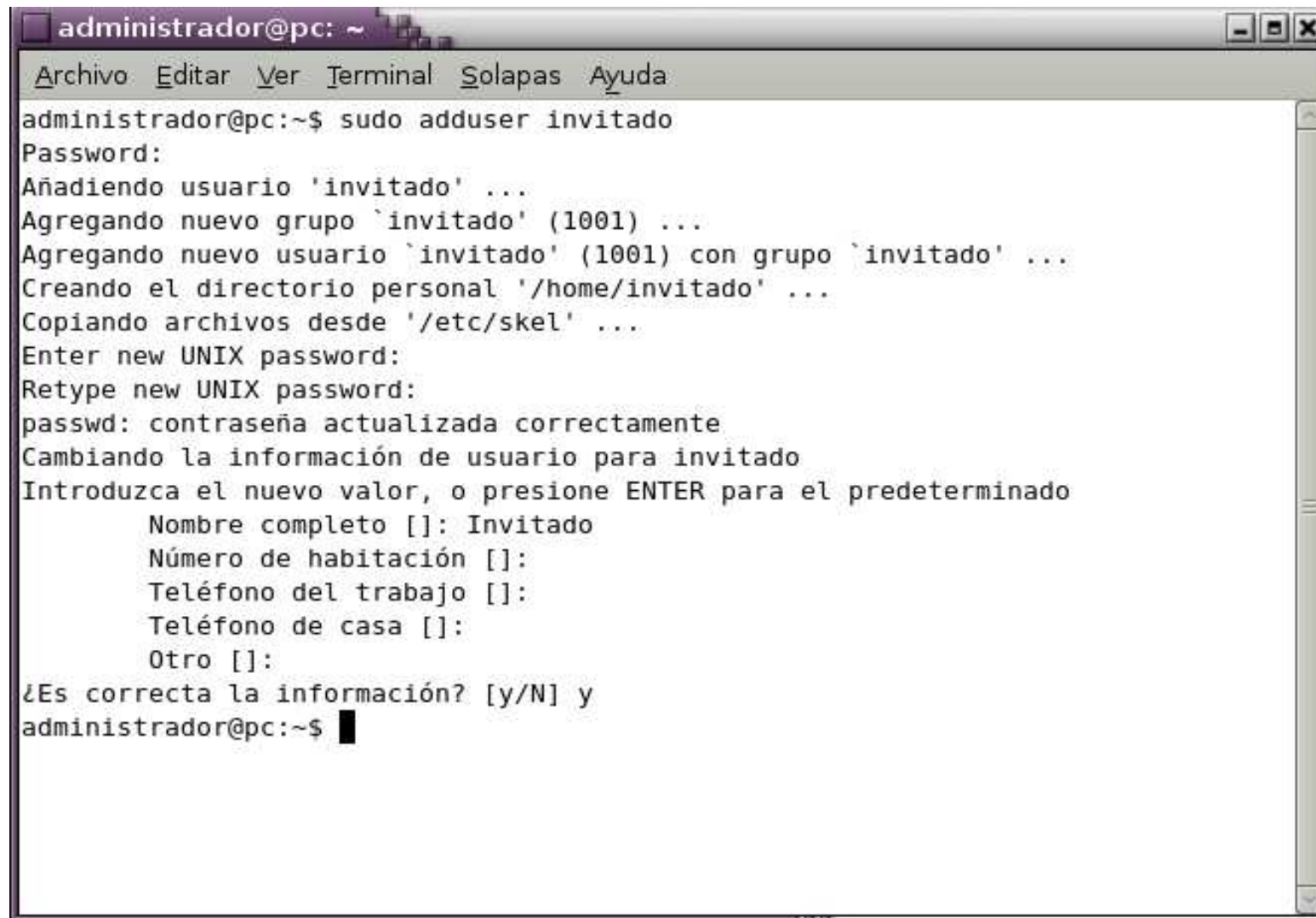
- **sudo.** Permite ejecutar comandos con permisos de superusuario (siempre que el usuario que lo use esté configurado en el fichero /etc/sudoers o sea del grupo del fichero):

**sudo comando**

# 1. USUARIOS Y GRUPOS. MODO TEXTO

- **adduser.** Permite crear usuarios en Linux. Su sintaxis es:  
**adduser [opciones] usuario [grupo]**
- Donde:
  - **--ingroup nombregrupo:** Crea los usuarios con grupo nombregrupo como principal
  - **--gid num\_gid:** igual que el anterior pero se indica el GID en lugar del nombre del grupo
  - **--home directorio:** hace que directorio sea el directorio personal del usuario (en lugar de /home/usuario)
- Si se utiliza grupo al final del comando, adduser no crea un nuevo usuario sino que añade uno existente (usuario) a un grupo existente (grupo) como grupo secundario

# 1. USUARIOS Y GRUPOS. MODO TEXTO



```
administrador@pc: ~  
Archivo Editar Ver Terminal Solapas Ayuda  
administrador@pc:~$ sudo adduser invitado  
Password:  
Añadiendo usuario 'invitado' ...  
Agregando nuevo grupo 'invitado' (1001) ...  
Agregando nuevo usuario 'invitado' (1001) con grupo 'invitado' ...  
Creando el directorio personal '/home/invitado' ...  
Copiando archivos desde '/etc/skel' ...  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: contraseña actualizada correctamente  
Cambiando la información de usuario para invitado  
Introduzca el nuevo valor, o presione ENTER para el predeterminado  
    Nombre completo []: Invitado  
    Número de habitación []:  
    Teléfono del trabajo []:  
    Teléfono de casa []:  
    Otro []:  
¿Es correcta la información? [y/N] y  
administrador@pc:~$
```

# 1. USUARIOS Y GRUPOS. MODO TEXTO

- **deluser.** Permite borrar cuentas de usuarios en Linux.  
`deluser [opciones] nombre_usuario`
- Donde:
  - Por defecto no borra el directorio ni los ficheros del usuario
  - `--remove-home`: borra la carpeta del usuario y su correo
  - `--remove-all-files`: borra todos los ficheros propiedad del usuario
  - `nombre_usuario` identifica al usuario que se va a borrar



# 1. USUARIOS Y GRUPOS. MODO TEXTO

- **usermod.** Se puede cambiar cualquier característica de la cuenta de un usuario editando directamente el fichero `/etc/passwd` o usando el comando `usermod`:

```
usermod [-c comentarios] [-d home_directory] [-m] [-e final de datos] [-g grupo inicial] [-G otros grupos [...]] [-l nombre de login] [-s Shell] [-u uid] [-o]] login
```

- Donde:
  - -c comentarios: modifica el comentario acerca del usuario
  - -d home\_directory: para especificar un directorio personal concreto, con -m mueve el contenido del antiguo
  - -e YYYY-MM-DD: fecha en la que la cuenta de usuario será desactivada
  - -g grupo inicial: nuevo número de grupo o nombre de grupo (existente)

# 1. USUARIOS Y GRUPOS. MODO TEXTO

```
usermod [-c comentarios] [-d home_directory] [-e final de datos] [-g  
grupo inicial] [-G otros grupos [,...]] [-l nombre de login] [-s Shell] [-u  
uid] [-o]] login
```

## ○ Donde:

- -G otros grupos: lista de grupos suplementarios de los que el usuario es miembro. Cada grupo se separa del siguiente por coma, sin espacio en blanco. Los grupos deben existir. Si el usuario es miembro de un grupo que no esté en esta lista, se borrará el grupo
- -l nombre de login: cambio de login
- -s shell: cambio de Shell
- -u uid: cambio de número de usuario
- login: actual del usuario que vamos a modificar

# 1. USUARIOS Y GRUPOS. MODO TEXTO

- Los grupos son agrupaciones de usuarios que cuentan con los mismos permisos, lo que hace más fácil el seguimiento y la administración del sistema
- Existen unos grupos especiales creados por el sistema, que engloban tareas específicas
- Algunas de estas cuentas son:
  - root (grupo de administradores)
  - users (grupo por defecto de los usuarios)
  - dialout (grupo que permite realizar conexiones remotas)
  - etc.

# 1. USUARIOS Y GRUPOS. MODO TEXTO

- **addgroup.** Permite crear grupos en Linux. Su sintaxis es:

`addgroup grupo`

- **delgroup.** Permite borrar grupos en Linux. Su sintaxis es:

`delgroup grupo`

- **groupmod.** Permite modificar grupos en Linux:

`groupmod [-g nuevo_gid] [-n nuevo_nombre] grupo`

- Donde:

- -n nuevo\_nombre especifica el nuevo nombre del grupo
- -g nuevo\_gid especifica el nuevo GID del grupo
- grupo nombre del grupo que se va a modificar

# 1. USUARIOS Y GRUPOS. MODO TEXTO

- **gpsswd.** Añade o elimina usuarios de un grupo.  
También se utiliza para añadir contraseñas a un grupo

gpsswd [opciones] usuario grupo

- Donde:
  - -a : Añade el usuario al grupo
  - -d: Elimina el usuario del grupo
  - -M: Añade varios usuarios al grupo

- **id.** Muestra información sobre el usuario, como el UID, el nombre de los grupos a los que pertenece y sus GID

id [opciones] [usuario]

- Donde:
  - -u | --user: escribe solo el UID del usuario
  - -n | --name: escribe el nombre del usuario

# 1. USUARIOS Y GRUPOS. MODO TEXTO

- **groups.** Para visualizar la lista de nombres de los grupos a los que pertenece un usuario

```
groups [usuario]
```

- **finger.** Muestra la información del usuario

```
finger [usuario]
```

- **newgrp.** Un usuario puede pertenecer a más de un grupo aunque sólo está activo uno y el usuario puede necesitar cambiarlo para crear nuevos ficheros

```
newgrp nuevo_grupo
```

# 1. USUARIOS Y GRUPOS. MODO TEXTO

- **chown.** Este comando cambia el propietario de un fichero. La sintaxis es:  
`chown [opciones] [propietario[:grupo]] fichero/s`
- Donde:
  - -r | --recursive: cambia el propietario y el grupo a un árbol de directorios
  - propietario: es el nombre o el identificador del nuevo propietario
  - grupo: si se desea cambiar también de grupo el fichero
  - fichero/s: listado de ficheros cuyo propietario se va a cambiar

# 1. USUARIOS Y GRUPOS. MODO TEXTO

- **chgrp**. Este comando cambia de grupo a un fichero. La sintaxis es:

```
chgrp [opciones] nuevo_grupo fichero/s
```

- Donde:
  - -r | --recursive: cambia el grupo a un árbol de directorios
  - nuevo\_grupo se sustituye por el nombre o el identificador de grupo del nuevo grupo



## 2. PERMISOS

- La seguridad de la información que almacenamos en los ordenadores centrales o servidores es fundamental, ya que muchos usuarios tendrán o podrán tener acceso a parte de los recursos sw y hw que están gestionados en estos ordenadores
- Cada usuario debe identificarse con un nombre y una password que le permitan conectarse al sistema
- Nunca deben utilizarse contraseñas (passwords) que puedan detectarse con facilidad, como nombres propios, aficiones, fechas significativas, etc

## 2. PERMISOS

- passwd. Permite al usuario cambiar o asignar una contraseña, siempre que el administrador lo autorice:

passwd

- Ejemplo:

\$ passwd

Passwd: changing password for user01

Old password:

New password:

Re-enter new password:

Password Changed

## 2. PERMISOS

- Al cambiar la palabra clave, hay que seguir una serie de *normas*. Estas normas suelen ser definidas por el administrador del sistema, y son las siguientes:
  - La palabra clave ha de ser diferente de la anterior
  - Debe tener al menos 6 caracteres
  - Al menos 2 caracteres tienen que ser alfabéticos
  - Debe tener al menos 1 carácter numérico o especial
  - Tiene que ser distinta del nombre de cuenta
  - No se pueden utilizar como palabra clave los mismos caracteres asignados como nombre de usuario cambiados de orden
  - No se puede cambiar la contraseña de mayúsculas a minúsculas

## 2. PERMISOS

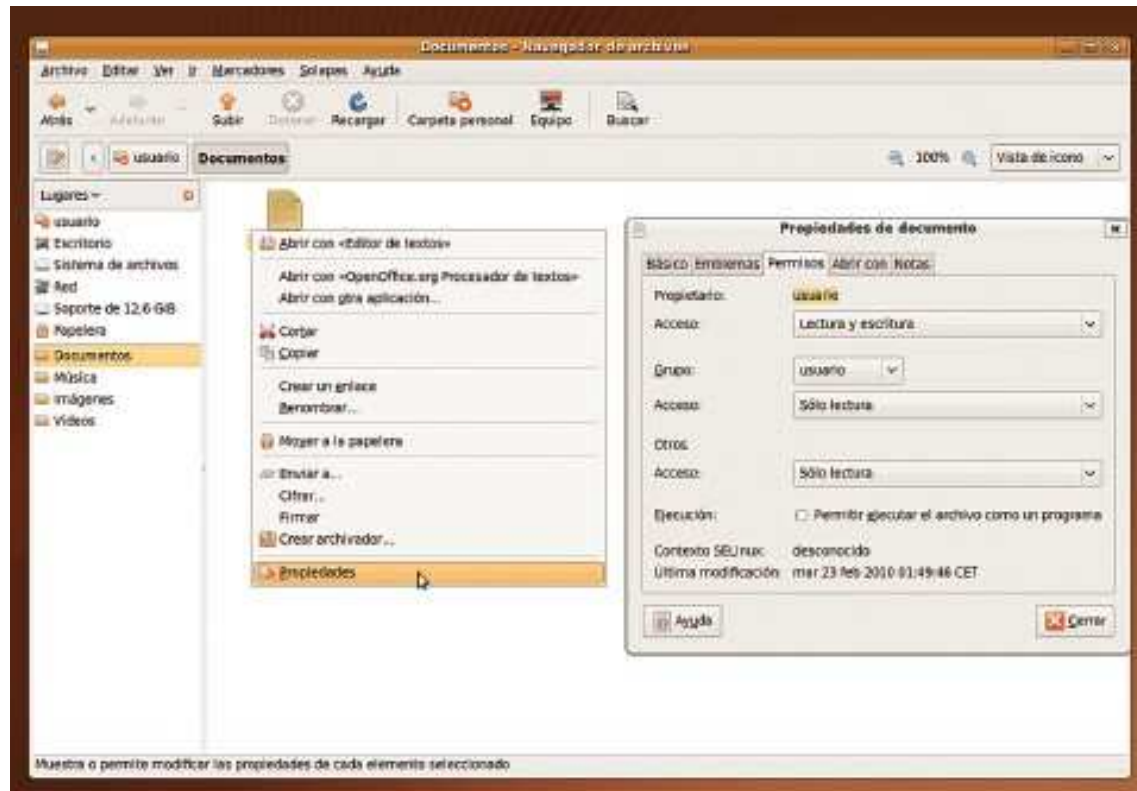
- Otra medida de seguridad es la protección de ficheros contra accesos no deseados, indicando qué usuarios pueden acceder y qué acciones pueden realizar sobre los datos
- Los permisos de acceso se establecen en tres niveles:
  - El del usuario que ha creado el fichero (propietario)
  - El del grupo al que pertenece el usuario (grupo)
  - El de todos los demás usuarios del sistema (otros)
- Para cada nivel hay tres tipos de acceso:
  - Lectura (r): Permite ver el contenido del fichero o directorio
  - Escritura (w): Permite modificar el contenido de un fichero, o crear y borrar ficheros y subdirectorios, si es un directorio
  - Ejecución (x): Se puede ejecutar el fichero o acceder a él si es un directorio

## 2. PERMISOS

- Los permisos para cada uno de los niveles aparecen siempre en el mismo orden: rwx.
- El carácter “r” significa acceso de lectura, “w” acceso de escritura, “x” acceso de ejecución y “-” que no tiene acceso de lectura, escritura y ejecución, dependiendo del lugar que ocupe
- Ejemplo: -rwxrw---x
  - Es un fichero de datos (-), con permiso de lectura, escritura y ejecución para el propietario (rwx), de lectura y escritura para el grupo (rw-) y de ejecución para otros (--x).
- Existe otra forma para referirnos a los permisos sobre ficheros: la máscara octal

## 2. PERMISOS. MODO GRÁFICO

- Sobre cualquier archivo podemos modificar sus permisos (si tenemos suficientes privilegios):
  - **Propiedades - Permisos**



## 2. PERMISOS

### ○ Máscara octal:

- Para poner la máscara de numeración octal se siguen los siguientes pasos:
- Se pasa la máscara a binario utilizando tres dígitos con valores 0 y/o 1, para cada nivel: propietario, grupo y otros. 0 significa que no tiene permiso y 1 que sí lo tiene.
- Las posibles combinaciones serían:

000 ---	100 r--
001 --x	101 r-x
010 -w-	110 rw-
011 -wx	111 rwx

## 2. PERMISOS

### ○ Máscara octal:

- A continuación se le asigna el valor 4, 2 y 1 a cada dígito binario
- Por ejemplo, para los permisos r-x, sería:
- Se suman los valores en cuya casilla haya un 1
- Siguiendo con el ejemplo anterior:  $4+1 = 5$
- La máscara octal para todos los permisos sería:

4	2	1
---	---	---

1	0	1
---	---	---

--- 0	r-- 4
--x 1	r-x 5
-w- 2	rw- 6
-wx 3	rwX 7



## 2. PERMISOS. MODO TEXTO

- **chmod.** Permite modificar los permisos de uno o varios ficheros. Su sintaxis es:

`chmod mascara fichero/s`

- donde la máscara puede ser octal o simbólica.
- Ejemplo:

`$ chmod 652 practica`

- El propietario de practica tendrá permiso de lectura y escritura ( $4+2=6$ ), el grupo de lectura y ejecución ( $4+1=5$ ) y otros de escritura (2)
- Si quisiéramos expresar lo mismo mediante una máscara simbólica, pondríamos:

`$ chmod u=rw, g=rx, o=w practica`

## 2. PERMISOS. MODO TEXTO

- La máscara simbólica está formada por tres códigos:

Clases de usuario	
Símbolo	Significado
u	propietario
g	grupo
o	otros
a	todos

Permisos	
Símbolo	Significado
r	lectura
w	escritura
x	ejecución

Operación	
Símbolo	Significado
+	añadir permisos
-	quitar permisos
=	asignar permisos

## 2. PERMISOS. MODO TEXTO

- umask. Cambia o muestra los permisos que se asigna por defecto. Su sintaxis es:  
`umask [mascara]`
- donde la máscara será octal
- Su funcionamiento es un tanto complicado, de ahí su poco uso actual:
  - Para los directorios hay que restarle el umask de los permisos básicos (777)
  - Para los directorios hay que hacer AND de la negación de umask con los permisos básicos(666)

### 3. CONCEPTO DE PROCESOS

- Un Proceso es un programa en ejecución
- PID: número de identificación que se asigna a cada proceso
- El propio Shell con el que está trabajando es un proceso lanzado por el sistema
- Los números de proceso se asignan secuencialmente cuando se generan nuevos procesos hijos
- El proceso 0 es el primero que se genera al principio de conectar con el sistema; el proceso 1 es el proceso *init*, a partir de cual se generan todos los restantes. El resto de procesos empezarán por el 2 como identificador. Lo normal es que no se lancen tantos procesos como para que el sistema no pueda ejecutarlos

## 4. GESTIÓN DE PROCESOS. ENTORNO GRÁFICO

- Monitor del Sistema:
  - Sistema – Administración – Monitor del sistema



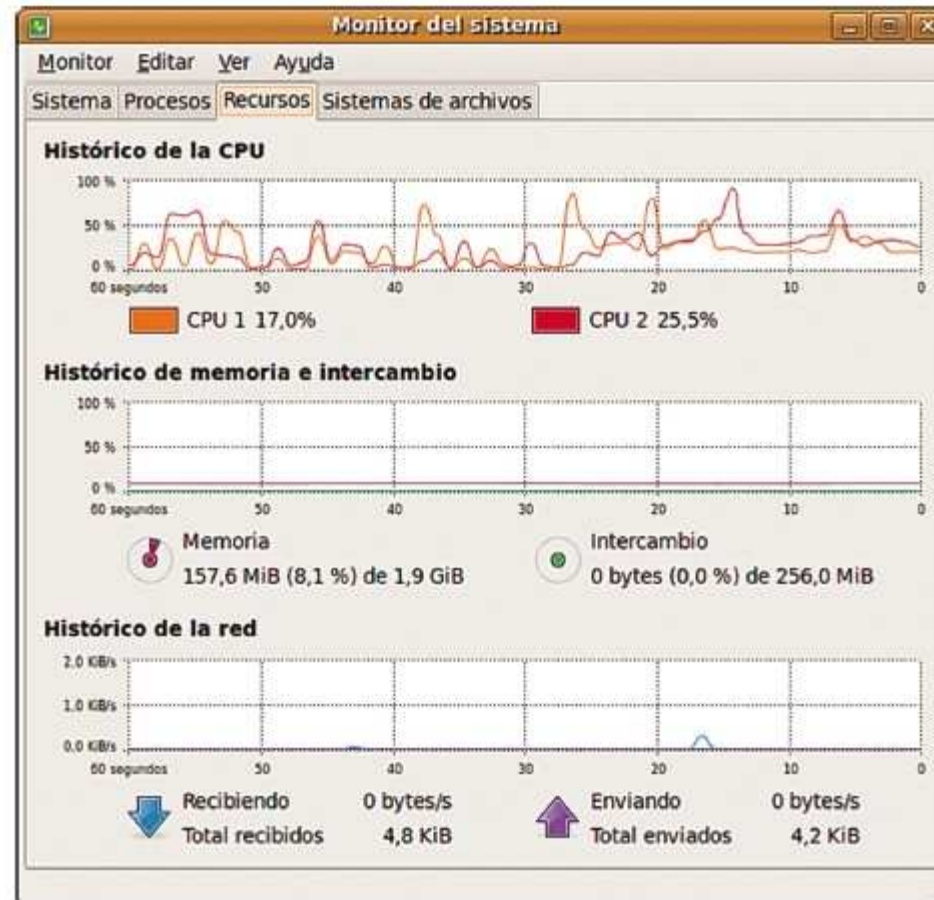
## ○ Monitor del Sistema: Procesos

Raspbian GNU/Linux 10 (buster)  
 kernel 4.19.0-10-arm64  
 64-bit ARMv8 processor  
 CARGA MEDIA para los últimos 1, 5 y 15 minutos: 0.27, 0.34, 0.23

Nombre del proceso	Usuario	Estado	% CPU	Prioridad	PID	Memoria
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-settings-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-system-monitor	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-keyring-daemon	usuario	Dormiendo	0	0	3248	328.0 KB
gnome-shell	usuario	Dormiendo	0	0	3248	328.0 KB

## 4. GESTIÓN DE PROCESOS. ENTORNO GRÁFICO

### ○ Monitor del Sistema: Recursos





## 4. GESTIÓN DE PROCESOS. MODO TEXTO

- **ps.** Muestra información sobre todos los procesos en ejecución. Su sintaxis es:

`ps [opciones]`

- Sin opciones: se muestran el PID de cada proceso activo del usuario, el terminal asociado, el tiempo CPU utilizado y el nombre de la orden que genera el proceso
- -f muestra una información más completa de cada proceso, con el usuario propietario del proceso, tiempo del comienzo de ejecución, etc.
- -l muestra toda información cada proceso con los códigos de cada campo
- -e muestra la información de todos los procesos activos del sistema, no sólo los del usuario
- -a muestra sólo los procesos asociados al terminal



## 4. GESTIÓN DE PROCESOS. MODO TEXTO

- **pstree.** Muestra los procesos en forma de árbol. ejecución. Su sintaxis es:

`pstree [opciones]`

- Donde:
  - -A: Muestra las líneas del árbol en estilo ASCII
  - -G: Muestra las líneas del árbol en estilo terminal VT100
  - -u: Muestra el propietario de los procesos

## 4. GESTIÓN DE PROCESOS. MODO TEXTO

- **kill:** finaliza su ejecución de un comando. Para terminar el proceso, es suficiente con su PID:
    - Ejemplo: \$ kill 201
  - Si queremos eliminar un proceso, sea el que sea, y de la forma anterior no se elimina, es debido a que el sistema no permite detener el proceso en ejecución. Si queremos terminarlo de todas formas, enviaremos el valor 9 como parámetro del comando kill. (kill -9 201)
  - Si lo que deseamos es terminar todos los procesos existentes en el sistema, ejecutaremos el comando kill con el valor 0
- 
- **sleep.** Este comando sirve para detener, por un intervalo de tiempo especificado en segundos, por defecto, un proceso determinado.

## 4. GESTIÓN DE PROCESOS. MODO TEXTO

- **wait.** Permite detener un proceso ejecutándose en segundo plano mientras se está ejecutando un proceso hijo

wait [pid]

- **time.** Calcula el tiempo que un proceso tarda en ejecutarse

time [opciones] [comando] [argumentos]

- **nohup.** Permite que un proceso permanezca vivo a pesar de desconectar el terminal (el núcleo del sistema operativo envía una señal a todos los procesos que estaban asociados a él para finalizarlos)

nohup [comando] [argumentos]

## 4. GESTIÓN DE PROCESOS. MODO TEXTO

- **nice**. Muestra o modifica la prioridad con la que se está ejecutando un proceso  
`nice [fichero]`
  - Si se pone sin opciones muestra la prioridad de la shell
  - Si se añade un fichero o proceso a continuación, se modifica su prioridad a un valor 10
  - -nNº: Aumenta en el Nº de veces la prioridad del proceso que se añada a continuación

## 4. GESTIÓN DE PROCESOS. MODO TEXTO

- Procesos en primer y en segundo plano:
  - Es posible ejecutar tanto proceso en tiempo real como en segundo plano, dándose prioridad a los procesos en tiempo real sobre cualquier otro tipo, exceptuando, claro está, a los procesos del sistema
  - En general, cuando un usuario está ejecutando un proceso, no puede lanzar otro hasta que este haya finalizado. Para solucionar este problema cada usuario puede ejecutar proceso en segundo plano o procesos background
- &. Ejecuta la orden precedente en segundo plano. El Shell devuelve el control al terminal mientras se ejecuta el proceso en segundo plano

Comando &

## 4. GESTIÓN DE PROCESOS. MODO TEXTO

- Variables de entorno relacionadas con los procesos
  - **\$:** Variable de entorno generada por el Shell que contiene el PID del proceso en ejecución. Visualizamos su valor utilizando `echo $$`
  - **!:** Variable de entorno generada por el Shell que contiene el PPID del proceso en ejecución (PID de su proceso padre). Visualizamos su valor utilizando `echo $!`
- Nota: Para abortar un proceso en ejecución usamos [Ctrl + C]

## 5. EJERCICIOS

1. Interpreta la siguiente información obtenida de unos archivos Linux:

```
drwxr-xr-x  3  user1  smr1  300 Apr 3 02:39  notas
lrw-r--r--   2  user2  smr1  366 Apr 1 12:30  cartas->commercial
-rwxrw-r--   1  user1  dao1  120 May 3 20:09  centrales
```

2. Ejercicio de aplicación 5.2
3. Ejercicio de aplicación 5.3
4. Ejercicio de aplicación 5.6
5. Ejercicio de aplicación 5.9
6. Ejercicio de aplicación 5.10

## 5. EJERCICIOS

7. Ejercicio de aplicación 5.11
8. Ejercicio de aplicación 5.12
9. Ejercicio de aplicación 5.13
10. Ejercicio de aplicación 5.14
11. Ejercicio de aplicación 5.15
12. Ejercicio de aplicación 5.16
13. Ejercicio de aplicación 5.17
14. Ejercicio de aplicación 5.18