

# UD 7. Gestión de usuarios y paquetes en Linux

Starting X display  
GNU/Linux 3.1  
m: \_

# **UD 7 – Gestión de usuarios y de paquetes en Linux**

## **INDICE DE CONTENIDOS**

**1.La gestión de usuarios**

**2.La instalación de paquetes de software**

# UD 7 – Gestión de usuarios y de paquetes en Linux

## INDICE DE CONTENIDOS

**1.La gestión de usuarios**

**2.La instalación de paquetes de software**

# UD 7 – Gestión de usuarios y de paquetes en Linux

## 1. La gestión de usuarios

Para Linux hay tres tipos de usuarios:

- Usuario root
- Usuarios normales o finales
- Usuario especiales o de sistema

**Root** es el “dueño” (owner) del sistema y tiene plenos poderes sobre él. También se le llama **administrador** o **superusuario**.

Es el único que puede administrar las cuentas de los usuarios.



# UD 7 – Gestión de usuarios y de paquetes en Linux

Los **usuarios normales o finales** son los que pueden utilizar las personas que se conectan al sistema. Este tipo de usuarios se caracteriza por:

- Normalmente poseen una **carpeta propia de trabajo** (home) en la que pueden crear sus propios archivos.
- Se **organizan por “grupos de usuarios”**.
- Un usuario siempre debe pertenecer, **al menos, a 1 grupo** de usuarios (su grupo primario)

Los **usuarios especiales o de sistema** son usuarios que crea el propio Linux para administrar sus “demonios” (servicios). Son usuarios que no pueden iniciar sesión y que son manejados por el sistema. **Ejemplos:** apache, daemon, bin, adm, lp, shutdown...



# UD 7 – Gestión de usuarios y de paquetes en Linux

## EL COMANDO SUDO

Cuando usamos usuarios finales (kike, pedro, juan...) no tenemos plenos poderes sobre el sistema, solo sobre los archivos de nuestra carpeta *home*.

Por otro lado, cuando instalamos un Ubuntu, nos pide que creemos nuestro usuario/contraseña inicial y el sistema nos crea un **usuario normal o final**.

Esto se hace así por razones de seguridad ante posibles ataques. Pero entonces **¿cómo puedo controlar el sistema si no soy root y nadie me ha dicho la contraseña del root?**



# UD 7 – Gestión de usuarios y de paquetes en Linux

En Linux no son tontos y te dejan un camino libre. Resulta que nuestro usuario normal pertenece a un **grupo especial llamado sudo** y lo podemos comprobar con el comando **groups** que nos dice a qué grupos pertenece nuestro usuario:

```
kike@kike-VirtualBox:/etc$ groups  
kike adm cdrom sudo dip plugdev lpadmin sambashare
```

Los usuarios miembros de este grupo (sudoers) se consideran usuarios de confianza para el administrador y **se les permite realizar acciones propias de root puntualmente.**

Ahora, si queremos hacer algo propio del root, simplemente anteponemos la palabra **sudo** y podremos hacerlo. Veamos un ejemplo:





# UD 7 – Gestión de usuarios y de paquetes en Linux

```
kike@kike-VirtualBox:/etc$ useradd pepito
useradd: Permission denied.
useradd: no se pudo bloquear /etc/passwd, inténtelo de nuevo.
kike@kike-VirtualBox:/etc$ sudo useradd pepito
[sudo] contraseña para kike:
kike@kike-VirtualBox:/etc$
```

**useradd** permite crear un usuario nuevo en el sistema, pero para realizarlo necesitamos privilegios de administrador.

Si anteponemos el comando **sudo**, nos pedirá nuestra contraseña (no la de root) y nos dejará hacerlo.

Esto tiene **muchas ventajas de seguridad**:

- root se usa lo menos posible (menos exposición)
- Se puede configurar qué permisos de administrador se le dan a los usuario *sudoers* (/etc/sudoers).
- El uso de **sudo** deja un rastro (log) que el administrador puede consultar





# UD 7 – Gestión de usuarios y de paquetes en Linux

**sudo** = SuperUsuario DOES “El superusuario hace”

Ahora sí podemos estudiar los comandos para la gestión de los usuarios del sistema. Veamos los principales:

- **users**: muestra los usuarios conectados actualmente
- **sudo adduser nombreusuario**: crea un nuevo usuario de nombre *nombreusuario*. También crea su carpeta *home* y además guarda más información (teléfono, despacho...).
- **sudo adduser --ingroup nombregrupo nombreusuario** : crea un nuevo usuario de nombre *nombreusuario* y hace que su grupo primario sea *nombregrupo*. También crea la carpeta *home*...
  - **man adduser**: para ver todas las opciones.



## UD 7 – Gestión de usuarios y de paquetes en Linux

- **sudo deluser nombreusuario**: borra el usuario de nombre *nombreusuario* pero sin borrar su carpeta *home*.
- **sudo deluser --remove-home nombreusuario**: igual que antes pero borrando la carpeta *home*.
  - **man deluser**: *para ver todas las opciones*
- **groups**: muestra los grupos a los que pertenece tu usuario
- **sudo addgroup nombregrupo**: crea un nuevo grupo de usuarios de nombre *nombregrupo*
- **sudo delgroup nombregrupo** : borra el grupo de usuarios de nombre *nombregrupo*



## UD 7 – Gestión de usuarios y de paquetes en Linux

- **passwd**: cambia la contraseña de tu usuario

```
kike@kike-VirtualBox:/home$ passwd
Cambiando la contraseña de kike.
(actual) contraseña de UNIX:
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
kike@kike-VirtualBox:/home$
```

- **sudo passwd nombreusuario**: cambia la contraseña del usuario de nombre *nombreusuario*.

```
kike@kike-VirtualBox:~$ sudo passwd pepito
[sudo] contraseña para kike:
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
kike@kike-VirtualBox:~$
```



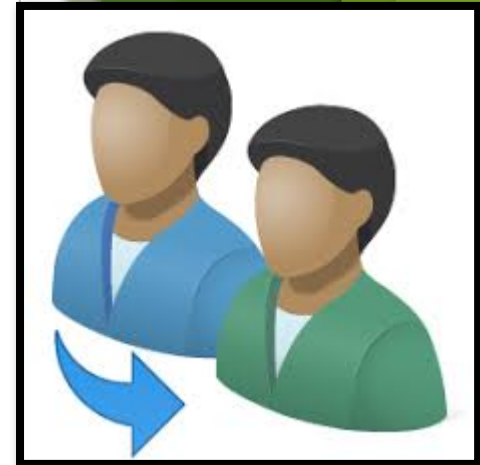
## UD 7 – Gestión de usuarios y de paquetes en Linux

- **su nombreusuario:** cambiar de usuario (Switch User). Permite iniciar sesión con otro usuario sin cerrar la sesión con tu usuario anterior. Te pide la contraseña del usuario destino. Ejemplo:

```
kike@kike-VirtualBox:~$ su pepito
Contraseña:
pepito@kike-VirtualBox:/home/kike$
```

- **sudo su nombreusuario:** cambio de usuario pero ahora solo te pide la contraseña del usuario *sudoer*, no la del destino.

```
kike@kike-VirtualBox:~$ sudo su pepito
[sudo] contraseña para kike:
pepito@kike-VirtualBox:/home/kike$ cd ..
pepito@kike-VirtualBox:/home$ cd ..
pepito@kike-VirtualBox:/$ exit
exit
kike@kike-VirtualBox:~$
```



# UD 7 – Gestión de usuarios y de paquetes en Linux

- **sudo su root**: nos cambiamos a root poniendo la contraseña del usuario sudoer. Otra forma abreviada es: **sudo su**

```
kike@kike-VirtualBox:~$ sudo su root  
[sudo] contraseña para kike:  
root@kike-VirtualBox:/home/kike#
```

- **sudo passwd root**: podemos cambiar la contraseña de root

```
kike@kike-VirtualBox:~$ sudo passwd root  
[sudo] contraseña para kike:  
Introduzca la nueva contraseña de UNIX:  
Vuelva a escribir la nueva contraseña de UNIX:  
passwd: contraseña actualizada correctamente  
kike@kike-VirtualBox:~$
```



En los ejercicios NO SE PUEDE USAR ROOT. Siempre se debe utilizar tu usuario normal y el comando sudo

# UD 7 – Gestión de usuarios y de paquetes en Linux

Ahora veamos **los comandos que permiten manipular** tanto el usuario como el grupo propietarios de un fichero.

**sudo *chown* nuevopropietario mifichero:** (CHange OWNer) este comando pone a *nuevopropietario* como el nuevo propietario del fichero *mifichero*. Si el usuario que lo ejecuta no lo hace con permisos de **sudo** no funciona.

**sudo *chgrp* nuevogrupo mifichero:** (CHange GRouP) este comando cambia el grupo propietario del fichero *mifichero* al grupo *nuevogrupo*. Si el usuario que lo ejecuta no lo hace con permisos de **sudo** no funciona.



# UD 7 – Gestión de usuarios y de paquetes en Linux

## INDICE DE CONTENIDOS

1.La gestión de usuarios

2.**La instalación de paquetes de software**



# UD 7 – Gestión de usuarios y de paquetes en Linux

## 2. La instalación de paquetes de software

Los paquetes de software pueden ser de dos tipos:

- **De código fuente:** puedes estudiar los ficheros y adaptarlos a tus necesidades pero también tienes que “compilarlos” tú mismo (nada sencillo)
- **De archivos binarios:** ya viene compilado para una familia de procesadores (Intel x86, ARM...) pero pierdes la versatilidad del código fuente.

En ambos casos los paquetes **dependen de otros** paquetes de códigos llamados “librerías” y que tienen utilidades de uso compartido por muchas aplicaciones (gestión de ventanas, funciones matemáticas...)



# UD 7 – Gestión de usuarios y de paquetes en Linux

Hacer la comprobación de dependencias entre paquetes “a mano” puede ser una tarea infernal. Para evitarlo se crearon los **gestores de paquetes de software**.

Los principales son:

- **Apt** y **apt-get** para distros Debian / Ubuntu
- **Yum**, **dnf** para RedHat, CentOS y Fedora

Funcionan desde la línea de comandos y su uso, en general, es similar. Necesitaremos **sudo** para poder usar cualquier gestor de paquetes.

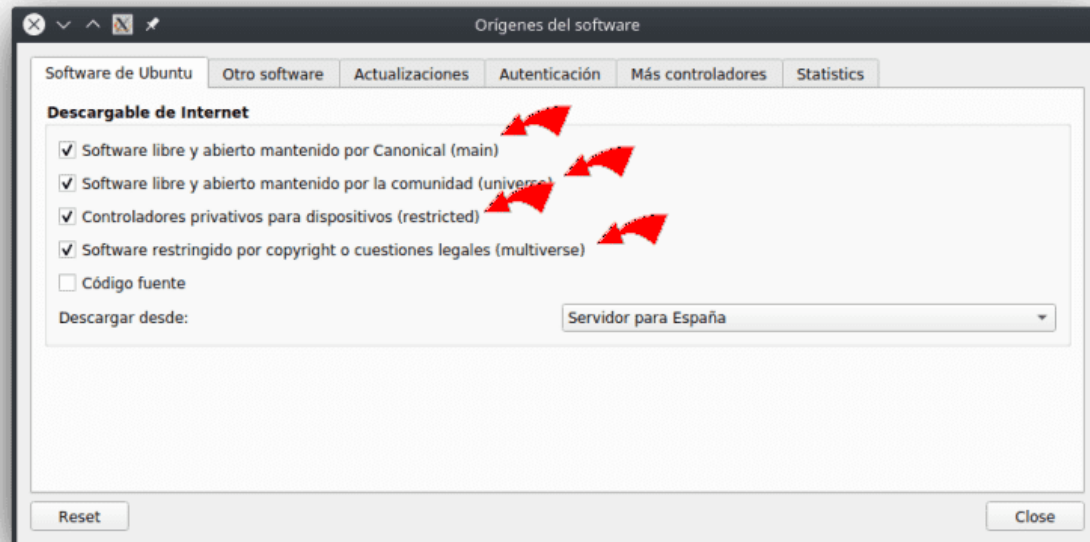
Nosotros estudiaremos **apt** (Advanced Package Tool) que además, comparte sintaxis con **apt-get**



# UD 7 – Gestión de usuarios y de paquetes en Linux

**apt** será nuestro instalador y desinstalador de software en Ubuntu. Se encargará de buscar tanto la última versión del paquete que queremos instalar como la última versión de las librerías que utilice.

Para poder hacer esto **apt** se conecta a distintos **orígenes o repositorios** de software.



# UD 7 – Gestión de usuarios y de paquetes en Linux

## Comandos apt para gestionar repositorios:

- **sudo apt add-repository *nombrerepositorio***: añade a la lista de repositorios activos el repositorio de nombre *nombrerepositorio*
- **sudo apt update**: busca actualizaciones pendientes de instalar en los repositorios activos
- **sudo apt upgrade**: instala las últimas actualizaciones del software que tengamos en nuestra máquina.

Estos dos últimos comandos se hacen a diario en los servidores de Internet.

apt-get update  
vs  
apt-get upgrade

# UD 7 – Gestión de usuarios y de paquetes en Linux

**Comandos apt para buscar software en los repositorios:**

- **apt-cache pkgnames *nombrepaquete*:** busca los paquetes cuyo nombre empiece por el texto *nombrepaquete*
- **apt-cache search *terminobusqueda*:** busca los paquetes que contengan el *terminobusqueda*.

No hace falta permisos de **sudo** ya que solo estamos consultando.

**Comandos para ver qué software tenemos instalado:**

- **apt list --installed**





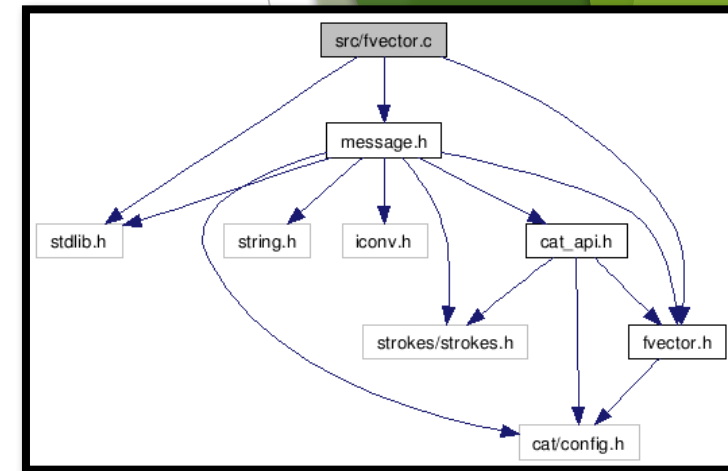
# UD 7 – Gestión de usuarios y de paquetes en Linux

## Comandos apt para instalar y desinstalar software:

- **sudo apt install *nombrepaquete***: instala tanto el paquete de software indicado como las dependencias requeridas, siempre que no estén previamente instaladas.
- **sudo apt remove *nombrepaquete***: desinstala el paquete de software indicado y aquellas dependencias que no sean usadas por otros paquetes instalados

## Comandos apt para “hacer limpieza”:

- **sudo apt autoremove**: desinstala dependencias obsoletas de software (versiones antiguas)



*Realizar los ejercicios  
de la sección  
Instalación de  
paquetes de software*