

EJERCICIO RESUELTO

Módulo: Implantación de Sistemas Operativos

Diferencias entre comandos Su y Sudo

Descripción:

Para realizar operaciones de configuración y administración en Linux se necesitan privilegios de root. Podemos llegar a ese estado de tener los permisos máximos para hacer cualquier cosa en el sistema de varias formas, con el comando SU y con el comando SUDO.

A continuación explicamos en que consiste la diferencia de entre estos dos comandos.

Objetivos:

- Usar debidamente los comandos de elevación de privilegios Su y Sudo

Recursos:

- Sistema operativo Linux.

Resolución:

Comandos Su y Sudo

En Linux existen dos comandos referentes al inicio de sesión con una cuenta de usuario y a iniciar sesión con derechos de administrador o root, veamos en qué consisten y cuando hay que usar cada uno de ellos.

Su

El comando **su** permite usar el intérprete de comandos de otro usuario sin necesidad de cerrar la sesión actual. Comúnmente se usa para obtener permisos de **root** para operaciones administrativas sin tener que salir y reentrar al sistema. Algunos entornos de escritorio, entre ellos GNOME y KDE, tienen programas que piden gráficamente una contraseña antes de permitir al usuario ejecutar un comando que usualmente requeriría tal acceso.

El nombre su proviene del inglés **s**ubstitute **u**ser (usuario sustituto). También hay quien lo hace derivar de **s**uper**u**ser (super-usuario, es decir, el usuario root o administrador) ya que habitualmente se utiliza para adoptar el rol de administrador del sistema.

Cuando se ejecuta, **su** pasando como parámetro otro nombre de usuario, se pide la contraseña de ese usuario, de esa forma iniciamos sesión como el usuario en cuestión.

```
Archivo Editar Ver Buscar Terminal Ayuda
amoreno@amoreno-VirtualBox:~$ su pepe
Contraseña:
pepe@amoreno-VirtualBox:/home/amoreno$ exit
exit
amoreno@amoreno-VirtualBox:~$
```

Una vez introducida la contraseña, podemos ejecutar los comandos como si fuésemos el otro usuario.

El directorio donde trabajamos no es el del nuevo usuario, es el directorio del último usuario usado.

```
Archivo Editar Ver Buscar Terminal Ayuda
amoreno@amoreno-VirtualBox:~$ pwd
/home/amoreno
amoreno@amoreno-VirtualBox:~$ su pepe
Contraseña:
pepe@amoreno-VirtualBox:/home/amoreno$ pwd
/home/amoreno
pepe@amoreno-VirtualBox:/home/amoreno$
```

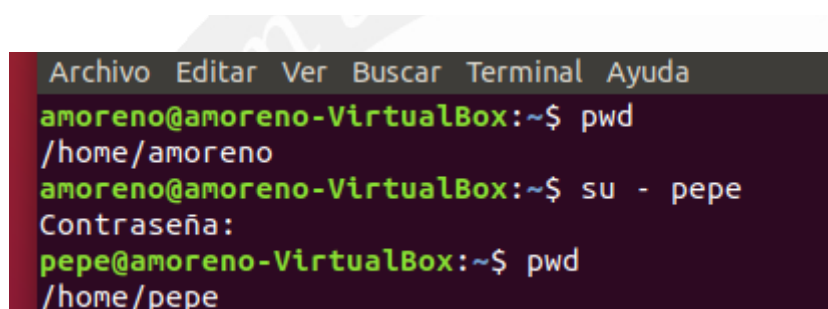
Al escribir **exit**, volvemos a nuestro usuario.

Una variante muy utilizada es usar **su** seguido de un guion y del nombre del usuario. Así, para logarse como el usuario pedro se tiene que teclear **su – pedro**

¿La diferencia entre usar o no el guion? **Se recomienda usar el guion porque** simula que inicias sesión con ese usuario; por consiguiente, ejecuta todos los archivos de inicio de ese usuario, cambia el directorio actual al HOME de ese usuario, cambia el valor de algunas variables del sistema adaptándolas al nuevo usuario (HOME, SHELL, TERM, USER, LOGNAME, entre otros).

Con **su** pepe (nos deja en el directorio del usuario anterior, no cambia de directorio al usuario nuevo)

Con **su – pepe** (cambia al directorio del usuario)



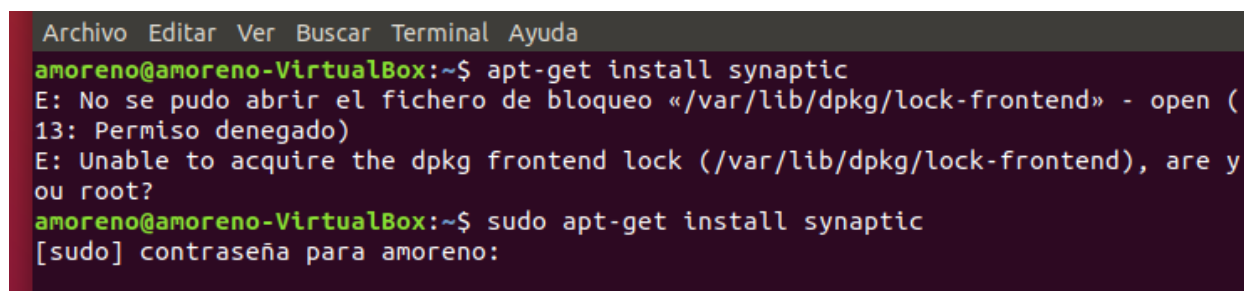
```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
amoreno@amoreno-VirtualBox:~$ pwd
/home/amoreno
amoreno@amoreno-VirtualBox:~$ su - pepe
Contraseña:
pepe@amoreno-VirtualBox:~$ pwd
/home/pepe

```

Sudo

Un comando relacionado, llamado **sudo**, ejecuta un comando como otro usuario, pero respetando una serie de restricciones sobre qué usuarios pueden ejecutar qué comandos en nombre de qué otros usuarios (usualmente especificadas en el archivo **/etc/sudoers**).



```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
amoreno@amoreno-VirtualBox:~$ apt-get install synaptic
E: No se pudo abrir el fichero de bloqueo «/var/lib/dpkg/lock-frontent» - open (
13: Permiso denegado)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), are y
ou root?
amoreno@amoreno-VirtualBox:~$ sudo apt-get install synaptic
[sudo] contraseña para amoreno:

```

Por otro lado, a diferencia de **su**, **sudo** pide a los usuarios su propia contraseña en lugar de la del usuario requerido; esto permite la delegación de comandos a usuarios en otras máquinas sin tener que compartir contraseñas, reduciendo el riesgo de dejar terminales desatendidas.

Problemas con sudo: el período de gracia

La ventaja de **sudo** respecto de **su** es que sólo ejecuta el comando solicitado simulando ser el otro usuario, sin cambiar verdaderamente el usuario actual. Ello implica que uno puede ejecutar un comando como administrador y, al segundo siguiente, volverá a tener los privilegios del usuario que estaba usando antes... o casi.

Algunos ven como una brecha de seguridad el hecho de que **sudo** otorgue un “período de gracia” que permita al usuario ejecutar comandos como otro usuario sin la necesidad de tener que teclear una y otra vez **sudo** delante del comando y la contraseña después de ejecutarlo. Pasado ese “período de gracia”, **sudo** volverá a preguntarnos la clave.

Esto no es recomendable, esencialmente porque alguien podría apoderarse de nuestro PC después de haber tecleado la contraseña de **sudo** y mientras el “período de gracia” está activo realizar cualquier operación con privilegios de root.

Afortunadamente, es posible deshabilitar el “período de gracia”, lo que va a mejorar la seguridad de tu sistema. Sólo hay que agregar una línea en el archivo **/etc/sudoers**:

```
sudo nano /etc/sudoers
```

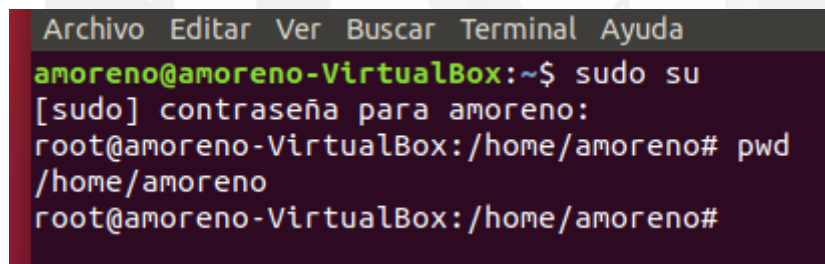
Y agregar la siguiente línea al final del archivo:

```
Defaults:ALL timestamp_timeout=0
```

Ese tiempo se cuenta en minutos (timeout)

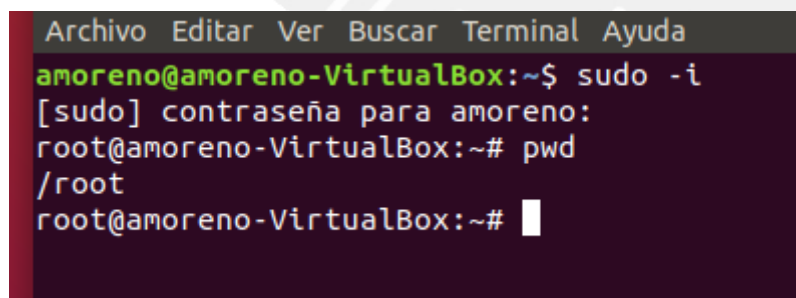
El cambio tiene efecto en forma inmediata, se guarda el fichero, sin necesidad de reiniciar el sistema.

Si usamos **sudo su**, da un tiempo de ejecución al usuario sudo de poder ejecutar una secuencia de instrucciones como usuario root, sin tener que ingresar una y otra vez la contraseña y luego ejecutar los comandos. Culminado ese periodo, sudo solicitará nuevamente la contraseña.



```
Archivo Editar Ver Buscar Terminal Ayuda
amoren@amoren-VirtualBox:~$ sudo su
[sudo] contraseña para amoren:
root@amoren-VirtualBox:/home/amoren# pwd
/home/amoren
root@amoren-VirtualBox:/home/amoren#
```

Si usamos **sudo -i**, simula que se inicia sesión con root; por consiguiente, ejecuta todos los archivos de inicio de ese root, cambia el directorio actual al HOME de root, cambia el valor de algunas variables del sistema adaptándolas a las variables de root (HOME, SHELL, TERM, USER, LOGNAME, entre otras)



```
Archivo Editar Ver Buscar Terminal Ayuda
amoren@amoren-VirtualBox:~$ sudo -i
[sudo] contraseña para amoren:
root@amoren-VirtualBox:~# pwd
/root
root@amoren-VirtualBox:~#
```