1 INSTALACIÓN DE MÁQUINA VIRTUAL Y UBUNTU

En primer lugar vamos a instalar un software de virtualización. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como «sistemas invitados», dentro de otro sistema operativo «anfitrión», cada uno con su propio ambiente virtual.

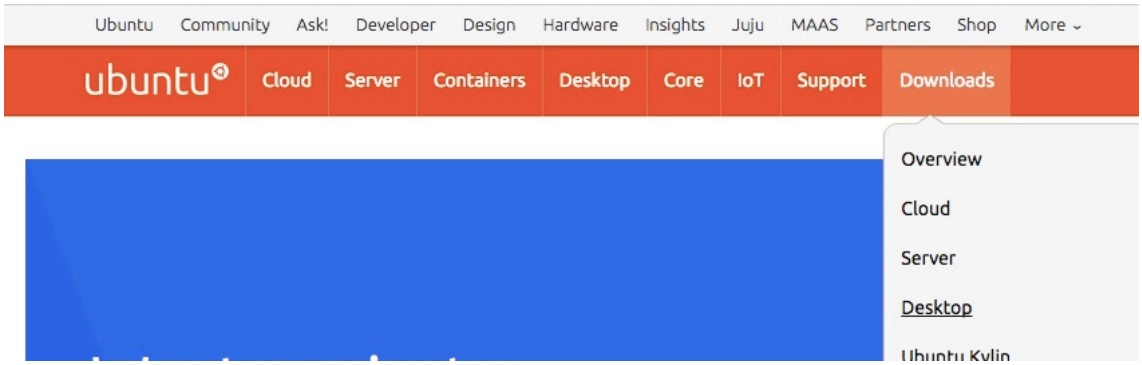
Lo descargamos de:

https://www.virtualbox.org/

Y procedemos a su instalación mediante el asistente.

A continuación vamos a descargar Ubuntu, la distribución más popular de GNU/Linux, de su web:

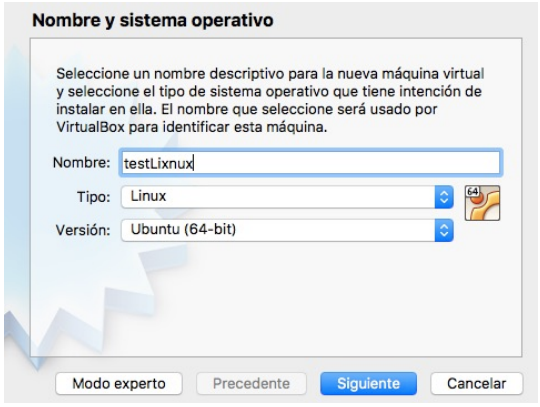
ubuntu.com



Y seleccionamos la opción más adecuada a nuestro SO anfitrión.

Iniciamos virtualbox

Seleccionamos nueva



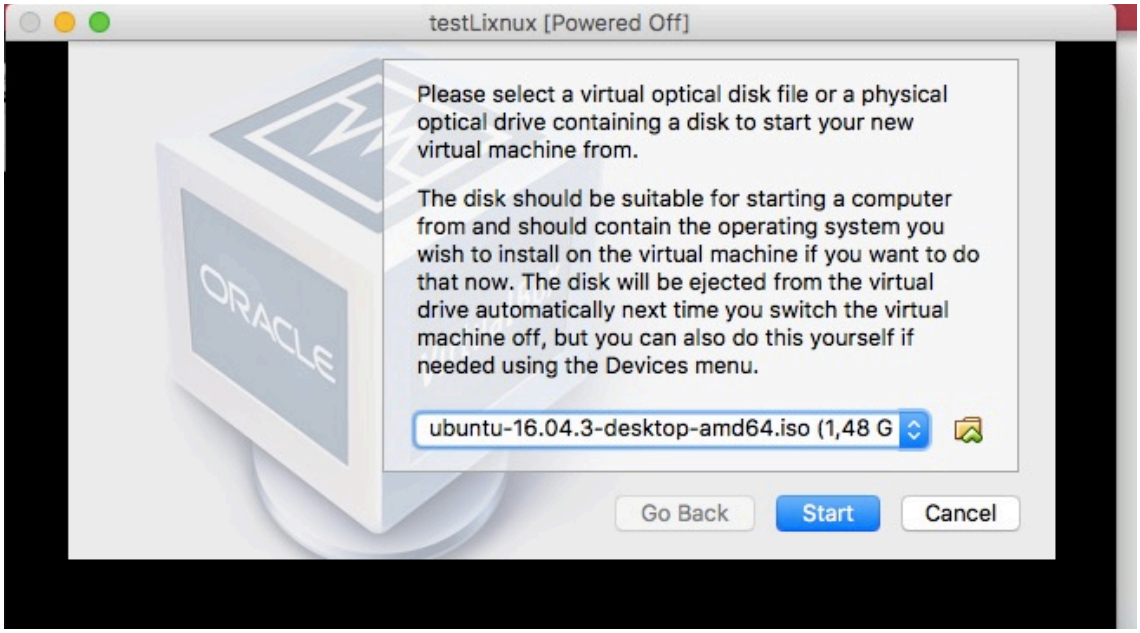
Seleccionamos la RAM

Creamos un disco duro virtual VDI

El almacenamiento de toda la máquina con reserva dinámica

Y unos 20 GB de almacenamiento de disco.

Una vez creada la máquina le damos a iniciar y seleccionamos donde tenemos la iso de Ubuntu que hemos descargado.



De esta manera directamente pulsamos en nuestro lenguaje y seleccionamos instalar Ubuntu.

Marcamos las dos casillas para instalar actualizaciones y firmware.

Marcamos borrar disco e instalar Ubuntu

Seleccionamos el resto de opciones nombre y usuario y contraseña.

2 CONEXIÓN A LINUX

Método local.

La conexión es tan sencilla como acceder a Ubuntu y trabajar en Linux a través de su terminal. Podemos lanzarla buscándola mediante el icono 'Buscar en el Equipo'. (La terminal se puede lanzar con CTRL+ALT+T).

También en local tenemos las denominadas TERMINALES VIRTUALES

Con ctrl izda+ alt + F1 se elimina el entorno gráfico de Ubuntu y aparece el entorno de texto siendo este caso una TT1, es decir una terminal 1 para poder trabajar.

Con la misma combinanción ctrl izda + alt + F1…F6 dispondremos de hasta 6 terminales virtuales disponibles.

Con ctrol izda + alt + F7 regresamos al entorno gráfico.

En remoto.

El servicio mediante SSH (secure Shell) podemos acceder de manera encriptada mediante la red. La arquitectura se lleva a cabo con un sistema cliente servidor en el que el cliente se conecta al puerto 22 del servidor.

Vamos a ver un ejemplo. Instalamos en primer lugar Open SSH, para ello en la terminal

sudo apt-get install openssh-server

Una vez instalado openssh para acceder de manera remota a una máquina podemos emplear el comando:

ssh nombreusuario@ip/hostmaquina

Por ejemplo en local:

ssh pedro@localhost

Lógicamente la el prompt de la terminal será el mismo pero estaremos accediendo desde la red local. Para salir escribimos logout

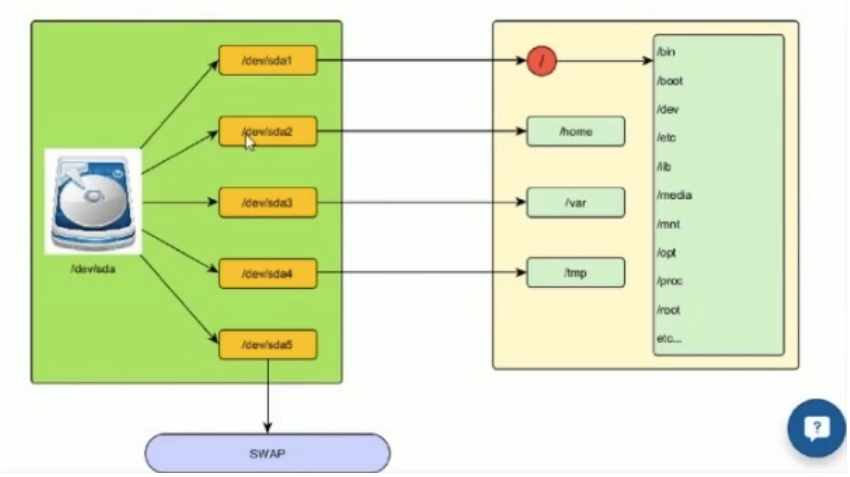
3 ESTRUCTURA DE DIRECTORIOS EN GNU/LINUX

La estructura de directorios de GNU/Linux se basa en el estándar FHS Filesystem Hierarchy Standard.

Sowftare del S.O.

/ => Raíz (nada que ver con las unidades de Windows)
|__ bin/ => Programas o ficheros ejecutables del sistema por todos los usuarios.
|__ sbin/ => Similar al anterior pero solo ejecutables por el admin.
|__ boot/ => Ficheros estáticos relacionados con el arranque (gestores de arranque)
|__ lib/ => Librerías compartidas.
|__ usr/ => Programas Linux. Puede incluir los subdirectorios lib y bin, o bien se se distribuyen en los anteriores. Por tanto una aplicación instalada en Linux se distribuye entre usr, bin y lib.
|__ opt => Software de terceros (no libre)
|__ dev/ => Sistema de ficheros virtuales para el uso de Hardware
|__ etc/ =>
|__ tmp/ => Ficheros temporales
|__ var/ => Directorio de datos variables
|__ home => Directorio de inicio de los diferentes usuarios.
| |__user/ => Directorio de cada uno de los usuarios.
|__ root => Directorio home del usuario root

Ejemplo de partición de un Disco duro en Linux:



4 SHELL

Cuando iniciamos la terminal en Linux, se inicia automáticamente una Shell. La Shell se compone de un prompt (encabezado) que se suele estar compuesto del nombre de usuario, nombre de la máquina o host y el directorio en el que nos encontramos, y un cursor, que es el que recibe los comandos.

Linux dispone de muchas Shell siendo la más utilizada BASH. Disponemos de atajos de teclado para emplear en la shell, así como comandos internos y variables de entornos.

Podemos comprobar la Shell utilizada con:

ps | grep $$

En su forma más habitual, el sistema operativo utiliza un signo de $ como prompt (introductor) para indicar que está preparado para aceptar ordenes o comandos, aunque este carácter puede ser fácilmente sustituido por otro u otros elegidos por el usuario.

En el caso de que el usuario acceda como usuario root este signo se sustituye por #.

La Shell tiene los siguientes atajos o ayudas de teclado:

atajo tab autocompleta los nombres y pulsando dos veces devuelve todas las ocurrencias.

atajo flecha arriba/abajo navega entre los comandos desde el inicio de sesión

atajo control izda se desplaza en la línea de comandos palabra por palabra

ctrl + R Nos permite buscar en el historial un comando determinado

5 COMANDOS BÁSICOS

Las ordenes de la Shell bash en Ubuntu siguen la secuencia

comando [-opciones] [argumentos]

y son sensibles a mayúsculas y minúsculas.

Comandos de ayuda (de los comandos)

Podemos emplear:

man nombredelcomando

o bien:

nombredelcomando --help

nombredelcomando –h

para una ayuda más resumida.

Visualización de archivos y elementos

ls muestra los directorios y archivos existentes en la ubicación

```
ls [opciones] [ruta]
```

ls –c ordenado por fecha de creación

ls –t ordenado por fecha de modificación

ls –l con detalles de cada archivo

ls –lsh con tamaño de cada archivo

ls –ld con propiedades de cada archivo

ls –a incluyendo archivos ocultos

ls –R de manera recursiva a los directorios que existan

ls | wc -l

tree similar al anterior en forma de árbol

file muestra las características de un archivo

```
file [OPCIÓN...] [ARCHIVO...]
```

Creación de directorios

mkdir ruta nombredeldirectorio

mkdir –p nombredirectorio1/nombredirectorio2 para crearlos recursivamente

mkdir –p nombredirectorio2/{nombredirectorio2,nombredirectorio3}

por ejemplo:

mkdir –p Música/blues/{80,90,00}

Borrado de directorios

rmdir borra directorios vacíos

rmdir –p borra subdirectorios

por ejemplo:

rmdir –p Música/blues/00

Navegación entre directorios

cd nos lleva al directorio home de usuario (sería lo mismo que cd ~/)

cd rutaabsoluta nos lleva a ese directorio

Por ejemplo:

cd /home/pedro/Música

cd / nos llevaría al directorio raíz

cd rutarelativa

Por ejemplo:

mkdir jazz

cd jazz

cd ../../Descargas

las rutas en Linux se separan por barra inclinada (no la contrabarra o barra invertida en Windows).

En las rutas relativas el uso de los dos puntos determina subir un nivel en el árbol de directorios, por ello, para subir un nivel ponemos:

cd ..

¿Para qué sirve el punto barra ./?

Cuando llamamos a archivos ejecutables que no se encuentran en las rutas de la variable PATH, necesitamos hacerlo incluyendo la ruta absoluta del archivo, aunque estemos en el directorio que lo contiene.

Para no tener que poner toda la ruta podemos poner ./ que hace referencia a la ruta absoluta en la que estamos.

Con cd y dos veces el tabulador podemos ir viendo los directorios que tenemos para navegar al deseado:

Por ejemplo:

cd pedro/ doble tabulador y nos muestra los directorios

Creación de archivos

touch crea un fichero vacío o cambia la fecha de uno existente

Por ejemplo:

touch holamundo.txt

echo sirve para introducir un mensaje.

Podemos usarlo para introducir un texto, por ejemplo:

echo ¡Hola Mundo! > holamundo.txt

Visualización de Archivos

cat muestra el contenido del fichero

cat holamundo.txt

Edición de Archivos

Podemos usar el editor nano

nano nombredelarchivo

Borrado de archivos

rm borra archivos

rm –r borra directorios y archivos de manera recursiva

rm –f no solicita confirmación para el borrado

rm –d borra directorios vacíos (similar a rmdir)

rm –v muestra el proceso de borrado

Copia de archivos y directorios

cp copia archivos y directorios

```
cp [opciones] ficheroorigen ficherodestino
```

```
cp [opciones] ficheroorigen directoriodestino
```

cd Imágenes

cp jovendelaperla.png ../Descargas/

Otro ejemplo sería copiar en nuestro home usando la virgulilla ~(altgr + ñ) como atajo del home de usuario:

cd Descargas

cp jovendelaperla.png ~/

Si por el contrario queremos copiar varios ficheros, habría que hacer lo siguiente:

touch log1.txt log2.txt log3.txt

mkdir logs

cp log1.txt log2.txt log3.txt /home/pedro/logs/

En caso de que queramos pasar todos los ficheros que se encuentran dentro de un directorio a otro, haremos lo siguiente:

cp /home/pedro/logs/ * /home/pedro/Descargas/

Para copiar un directorio entero (con sus archivos) empleamos

cp –R /home/pedro/logs/ /home/pedro/Descargas/

Como se puede ver, el comando cp en Linux ofrece muchas posibilidades. Hasta ahora hemos visto los comandos más habituales sin introducir las [opciones] o "flags" que hemos explicado en la sintaxis.

A continuación veremos los flags más utilizados con el comando cp en Linux:

Backup: -b

Crea un backup en el destino en el caso en el que exista un archivo llamado igual que el que queremos generar.

Force: -f

Fuerza el borrado de los archivos destino sin consultar o avisar al usuario.

Interactive: -i

Informa antes de sobrescribir un archivo en el destino indicado.

Link: -l

Realiza un link en vez de copiar los ficheros.

Preserve: -p

Realiza la copia de los ficheros y directorios conservando la fecha de modificación de los archivos y carpetas originales.

Copia recursiva: -r

Copia de forma recursiva.

Suffix: -S SUFFIX

Añade la palabra "SUFFIX" (o la palabra que le indiquemos, por ejemplo BACKUP) a los archivos de backup creados con el flag "–b".

Update: -u

El comando cp en Linux no copia un archivo o directorio a un destino si este destino tiene la misma fecha de modificación o una fecha de modificación posterior comparándola con el archivo o directorio que queremos mover.

Verbose: -v

Muestra lo que se está ejecutando.

Mover archivos y directorios

mv similar al anterior para mover archivos o directorios (con la mismas opciones)

Para renombrar usamos también mv, con la sintaxis:

mv archivo archivorenombrado

mv directorio directoriorenombrado

Enlaces simbólicos

Los enlaces simbólicos se pueden definir como los accesos directos en Sistemas Operativos Windows, es decir, su ejecución redirige a otros archivos o directorios.

Para crear un enlace simbólico empleamos

ln –s nombredirectorio nombrenlace

Por ejemplo, en home creamos:

ln –s Música/jazz jz

y listamos con ls –l para comprobar el enlace.

Podemos acceder al enlace simbólico jazz con cd y comprobar que si creamos un archivo, realmente lo estaremos haciendo en el directorio jazz

Nota: Para eliminar un enlace simbólico emplearemos rm no rmdir.

Búsqueda de Ficheros

find

```
find [/directorio/donde/buscar...] [-expresión] [búsqueda]
```

Donde "expresión" es el tipo de búsqueda y siempre se le antepone el signo "-"

La expresión "-name" sería para realizar una búsqueda por nombre.

Por ejemplo, para buscar en todo el sistema de archivos o raíz "/" las carpetas y archivos que se llamen "log.txt”:

find ~/ -name log.txt

Podemos usar asterisco:

find ~/ -name *og*

Otra expresión sería "-size" para realizar la búsqueda por tamaño. Por ejemplo podemos decirle que encuentre los archivos/carpetas de más de 1500 KB:

find ~/ -size +1500

La opción "2>/dev/null" es muy interesante para que no muestre los errores de "Permiso denegado". Por ejemplo para buscar en la raíz "/" el archivo "log1.txt":

find / -name log1.txt 2>/dev/null

grep (localizar) El comando grep localiza una palabra, clave o frase en un conjunto de directorios, indicando en cuáles de ellos la ha encontrado.

Este comando rastrea fichero por fichero, por turno, imprimiendo aquellas líneas que contienen el conjunto de caracteres buscado. Si el conjunto de caracteres a buscar está compuesto por dos o más palabras separadas por un espacio, se colocará el conjunto de caracteres entre comillas simples (').

La sinapsis del comando sería:

```
grep [OPCIÓN] 'conjuntocaracteres' file1 file2 file3
```

siendo 'conjuntocaracteres' la secuencia de caracteres a buscar, y file1, file2, y file3 los ficheros donde se debe buscar.

Por ejemplo:

touch log1.txt => Escribimos Lorem ipsum…

touch log2.txt => Escribimos En un lugar de la Mancha…

grep 'Mancha' log1.txt log2.txt *.* para todos los archivos

Las opciones principales del comando son:

-c → lo único que se hace es escribir el número de las líneas que satisfacen la condición.
-i → no se distinguen mayúsculas y minúsculas.
-l → se escriben los nombres de los ficheros que contienen líneas buscadas.
-n → cada línea es precedida por su número en el fichero.
-s → no se vuelcan los mensajes que indican que un fichero no se puede abrir.
-v → se muestran sólo las líneas que no satisfacen el criterio de selección.

Varios

pwd nos devuelve la ruta desde el directorio raíz

clear limpia la Shell

reset resetea la terminal

history muestra el historial de comandos

history -c borra el historial de comandos

hostname nombre de la máquina

who usuarios y terminales virtual

whoami qué usuario soy

date fecha

cal calendario

logout cierra sesión

reboot reinicia el equipo

halt cierra el equipo (exige sudo)

poweroff ídem

shutdown opciones ídem, por ejemplo:

sudo shutdown -r now

sudo shutdown -r +5

sudo shudown -r 22:30

Comandos Internos (no generan PID)

alias permite establecer un alias de un comando que tenga varias opciones para escribirlo más rápido. La sintaxis para generar nuevos alias es:

alias nombrecomando = 'comando opciones'

El comando va entre comillas simples. Pero la pregunta es ¿Dónde ponemos esto? Pues si queremos que solo sea temporal, simplemente lo escribimos en la consola y durará hasta que la cerremos.

Ahora, si lo queremos de forma permanente, esto lo ponemos dentro del fichero ~/.bashrc el cual está en nuestro /home, y si no está, pues lo creamos (siempre con el punto delante).

Si escribimos alias nos mostrará todos los alias existentes.

echo muestra el valor de lo que pongamos a continuación

exec ejecuta un programa con el mismo PID de la consola, con lo cual al finalizar este programa finaliza también la consola.

env muestra todas las variables de entorno.

Este comando se puede emplear para cambiar el valor de una variable en una aplicación con la sintaxis:

env VARIABLE=nuevovalor binarioprograma

export crea una variable local

export VARIABLE='valor'
echo $VARIABLE

Variable de entorno PATH

Path es una variable de entorno para establecer la ruta de un ejecutable dentro del sistema, de tal manera que los ejecutables que estén en las rutas definidas en esta variable puedan ser ejecutados desde cualquier directorio del sistema.

La forma de almacenar las diferentes rutas en path es separándolas por dos puntos :.

Para ver la variable PATH escribimos:

printenv PATH

Si queremos ver donde se encuentra un ejecutable podemos emplear:

which ejecutable

y nos devolverá su ruta. Si esta se encuentra en PATH lo podremos ejecutar desde cualquier punto.

Vamos a crear un ejecutable y añadimos su ruta a PATH

mkdir misbin

touch holamundo

nano holamundo

Y escribimos:

echo “¡Hola Mundo!”

Guardamos y salimos y le cambiamos los permisos con

chmod 777 holamundo

Y lo ejecutamos de manera local con

./holamundo

Para introducir la ruta de este ejecutable en la variable PATH emplearemos:

export PATH=$PATH:/home/pedro/misbin

Ahora comprobamos como se puede ejecutar el archivo desde cualquier ubicación simplemente con:

holamundo

6 GESTIÓN DE USUARIOS

Linux es un sistema multiusuario, por lo tanto, la tarea de añadir, modificar, eliminar y en general administrar usuarios se convierte en algo no solo rutinario, sino importante, además de ser un elemento de seguridad que mal administrado o tomado a la ligera, puede convertirse en un enorme hoyo de seguridad.

Los usuarios en Unix/Linux se identifican por un número único de usuario, User ID, UID. Y pertenecen a un grupo principal de usuario, identificado también

por un número único de grupo, Group ID, GID. El usuario puede pertenecer a más grupos además del principal.

Aunque sujeto a cierta polémica, es posible identificar tres tipos de usuarios en Linux:

Usuario root, también llamado superusuario o administrador. Se caracteriza por:

- Su UID (User ID) es 0 (cero).
- Es la única cuenta de usuario con privilegios sobre todo el sistema.
- Acceso total a todos los archivos y directorios con independencia de propietarios y permisos.
- Controla la administración de cuentas de usuarios.
- Ejecuta tareas de mantenimiento del sistema.
- Puede detener el sistema.
- Instala software en el sistema.
- Puede modificar o reconfigurar el kernel, controladores, etc.

Usuarios especiales. Por ejemplos bin, daemon, adm, lp, sync, shutdown, mail, operator, squid, apache, etc. Se caracterizan por:

- Se les llama también cuentas del sistema.
- No tiene todos los privilegios del usuario root, pero dependiendo de la cuenta asumen distintos privilegios de root.
- Lo anterior para proteger al sistema de posibles formas de vulnerar la seguridad.
- No tienen contraseñas pues son cuentas que no están diseñadas para iniciar sesiones con ellas.
- También se les conoce como cuentas de "no inicio de sesión" (nologin).
- Se crean (generalmente) automáticamente al momento de la instalación de Linux o de la aplicación.
- Generalmente se les asigna un UID entre 1 y 100 (definido en /etc/login.defs)

Usuarios normales. Se usan para usuarios individuales y se caracterizan por:

- Cada usuario dispone de un directorio de trabajo, ubicado generalmente en /home.
- Cada usuario puede personalizar su entorno de trabajo.
- Tienen solo privilegios completos en su directorio de trabajo o home.

- Por seguridad, es siempre mejor trabajar como un usuario normal en vez del usuario root, y cuando se requiera hacer uso de comandos solo de root, utilizar el comando su.
- En las distros actuales de Linux se les asigna generalmente un UID superior a 500.

Superusuario en Ubuntu

El superusuario o root tiene todos los permisos por lo que podríamos loguearnos como superusuario para poder realizar todas las operaciones que necesitemos.

Podemos hacerlo, pero lo normal es no logueranos como usuario root, sino emplear nuestro usuario (con el que hemos realizado la instalación) y preceder las instrucciones con el comando sudo (de super user do).

La primera vez que empleemos sudo en cada sesión de la terminal nos solicitará nuestra contraseña de usuario.

Si por algún motivo especial queremos emplear el usuario root en primer lugar debemos establecer su contraseña, para lo cual escribimos:

sudo passwd
>introducimos la contraseña de nuestro usuario
>introducimos la nueva contraseña de root
>confirmamos la nueva contraseña de root

Ahora para iniciar sesión como usuario root tecleamos

su root

y tecleamos su contraseña

Para salir de root usamos

exit

Si queremos deshabilitar root (mejor dicho su contraseña) empleamos:

sudo passwd -dl root

Otra opción en Ubuntu para trabajar como root es utilizar el comando:

sudo bash

Para ver qué usuarios tienen permisos de administrador podemos ver el archivo sudoers en el directorio etc

sudo cat sudoers

Creación de usuarios.

adduser es el comando más sencillo para añadir nuevos usuarios al sistema desde la línea de comandos.

Ahora bien, realmente no hay prácticamente necesidad de indicar ninguna opción ya que si hacemos lo siguiente:

sudo adduser luis

E introducimos su contraseña.

Se creará el usuario y su grupo (con el mismo nombre), así como las entradas correspondientes en /etc/passwd, /etc/shadow y /etc/group.

También se creará el directorio de inicio o de trabajo: /home/luis y los archivos de configuración que van dentro de este directorio y que más adelante se detallan.

Nota: En realidad adduser es un enlace simbólico a useradd, comando de bajo nivel para crear usuarios. Si usamos useradd, debemos añadir la opción –m para que cree su directorio en home y añadir a posteriori su contraseña con passwd. (para ver todas las opciones de useradd, teclear man useradd).

Para cambiar de usuario desde la terminal usamos:

su Luis

y vemos cómo cambia el prompt, aunque en el entorno gráfico el usuario se mantiene.

Para ver todos los usuarios podemos ver el archivo:

cat /etc/passwd

De la misma forma podemos ver los grupos con:

cat /etc/group

Nota: Una forma sencilla para ver todos los usuarios es lanzar un ls –l a home:

cd /home/
ls –l

la tercera y cuarta columna del listado serán el usuario propietario del directorio y el grupo.

Para saber el usuario de un grupo, usamos groups, por ejemplo:

groups luis

Crear usuario con otro grupo

sudo adduser nombreusuario --ingroup nombregrupo

Y para crear grupos usamos:

sudo addgroup nombregrupo

por ejemplo:

sudo addgroup usuarios

Por tanto:

sudo adduser lucia --ingroup usuarios

Cambiar contraseñas con passwd

sudo passwd luis

Las fechas de expiración de contraseña, etc. Quedan lo más amplias posibles así que no hay problema que la cuenta caduque.

El superusuario es el único que puede indicar el cambio o asignación de contraseñas de cualquier usuario. Los usuarios normales pueden cambiar su

contraseña en cualquier momento con tan solo invocar passwd sin argumentos, y podrá de esta manera cambiar la contraseña cuantas veces lo requiera, pero solo de sí mismo.

passwd tiene integrado validación de contraseñas comunes, cortas, de diccionario, etc.

Modificación de usuarios

usermod permite modificar o actualizar un usuario o cuenta ya existente. Sus opciones más comunes o importantes son las siguientes:

-c añade o modifica el comentario
-d modifica el directorio de trabajo o home del usuario
-e cambia o establece la fecha de expiración de la cuenta, formato AAAA-MM-DD
-g cambia el número de grupo principal del usuario (GID)
-G establece otros grupos a los que puede pertenecer el usuario
-l cambia el login o nombre del usuario
-L bloque la cuenta del usuario, no permitiéndole que ingrese al sistema. No borra ni cambia nada del usuario, solo lo deshabilita.
-s cambia el shell por defecto del usuario cuando ingrese al sistema.
-u cambia el UID del usuario.
-U desbloquea una cuenta previamente bloqueada con la opción -L.

Por ejemplo podemos cambiar el nombre de un usuario:

usermod luis –l luigi

Cambiar su directorio home;

mkdir usuarioluigi
usermod luigi –d usuarioluigi

Cambiar usuario de grupo

sudo usermod -g usuarios luis

Si quisiéramos que tuviera permisos de superusuario lo añadimos al grupo sudo:

sudo usermod –g sudo luis

Para modificar el nombre de un grupo usamos:

sudo groupmod -n users usuarios

Eliminación de usuarios

Como su nombre lo indica, userdel elimina una cuenta del sistema, userdel puede ser invocado de tres maneras:

userdel luis

Sin opciones elimina la cuenta del usuario de /etc/passwd y de /etc/shadow, pero no elimina su directorio de trabajo ni archivos contenidos en el mismo, esta es la mejor opción, ya que elimina la cuenta pero no la información de la misma.

userdel -r luis

Al igual que lo anterior elimina la cuenta totalmente, pero con la opción -r además elimina su directorio de trabajo y archivos y directorios contenidos en el mismo, así como su buzón de correo, si es que estuvieran configuradas las opciones de correo. La cuenta no se podrá eliminar si el usuario esta logueado o en el sistema al momento de ejecutar el comando.

userdel -f luis

La opción -f es igual que la opción -r, elimina todo lo del usuario, cuenta, directorios y archivos del usuario, pero además lo hace sin importar si el usuario está actualmente en el sistema trabajando. Es una opción muy radical, además de que podría causar inestabilidad en el sistema, así que hay que usarla solo en casos muy extremos.

Para eliminar un grupo empleamos:

delgroup nombredelgrupo

Cualquiera que sea el tipo de usuario, todas las cuentas se encuentran definidas en el archivo de configuración 'passwd', ubicado dentro del directorio /etc. Este archivo es de texto tipo ASCII, se crea al momento de la instalación con el usuario root y las cuentas especiales, más las cuentas de usuarios normales que se hayan indicado al momento de la instalación.

El archivo /etc/passwd contiene una línea para cada usuario, similar a las siguientes:

root:x:0:0:root:/root:/bin/bash
luis:x:501:500:Luis:/home/luis:/bin/bash

La información de cada usuario está dividida en 7 campos delimitados cada uno por ':' dos puntos.

/etc/passwd	
Campo 1	Es el nombre del usuario, identificador de inicio de sesión (login). Tiene que ser único.
Campo 2	La 'x' indica la contraseña encriptada del usuario, además también indica que se está haciendo uso del archivo **/etc/shadow**, si no se hace uso de este archivo, este campo se vería algo así como: '**ghy675gjuXCc12r5gt78uuu6R**'.
Campo 3	Número de identificación del usuario (UID). Tiene que ser único. 0 para root, generalmente las cuentas o usuarios especiales se numeran del 1 al 100 y las de usuario normal del 101 en delante, en las distribuciones mas recientes esta numeración comienza a partir del 500.
Campo 4	Numeración de identificación del grupo (GID). El que aparece es el número de grupo principal del usuario, pero puede pertenecer a otros, esto se configura en **/etc/groups**.
Campo 5	Comentarios o el nombre completo del usuario.
Campo 6	Directorio de trabajo (Home) donde se sitúa al usuario después del inicio de sesión.
Campo 7	Shell que va a utilizar el usuario de forma predeterminada.

Anteriormente (en sistemas Unix) las contraseñas cifradas se almacenaban en el mismo /etc/passwd. El problema es que 'passwd' es un archivo que puede ser leído por cualquier usuario del sistema, aunque solo puede ser modificado por root.

Con cualquier computadora potente de hoy en día, un buen programa de descifrado de contraseñas y paciencia es posible "crackear" contraseñas débiles (por eso la conveniencia de cambiar periódicamente la contraseña de root y de otras cuentas importantes). El archivo 'shadow', resuelve el problema ya que solo puede ser leido por root. Considérese a 'shadow' como una extensión de 'passwd' ya que no solo almacena la contraseña encriptada, sino que tiene otros campos de control de contraseñas.

El archivo /etc/shadow contiene una línea para cada usuario, similar a las siguientes:

root:ghy675gjuXCc12r5gt78uuu6R:10568:0:99999:7:7:-1::
sergio:rfgf886DG778sDFFDRRu78asd:10568:0:-1:9:-1:-1::

La información de cada usuario está dividida en 9 campos delimitados cada uno por ':' dos puntos.

/etc/shadow	
Campo 1	Nombre de la cuenta del usuario.
Campo 2	Contraseña cifrada o encriptada, un '*' indica cuenta de 'nologin'.
Campo 3	Días transcurridos desde el 1/ene/1970 hasta la fecha en que la contraseña fue cambiada por última vez.
Campo 4	Número de días que deben transcurrir hasta que la contraseña se pueda volver a cambiar.
Campo 5	Número de días tras los cuales hay que cambiar la contraseña. (-1 significa nunca). A partir de este dato se obtiene la fecha de expiración de la contraseña.
Campo 6	Número de días antes de la expiración de la contraseña en que se le avisará al usuario al inicio de la sesión.
Campo 7	Días después de la expiración en que la contraseña se inhabilitara, si es que no se cambio.
Campo 8	Fecha de caducidad de la cuenta. Se expresa en días transcurridos desde el 1/Enero/1970 (epoch).
Campo 9	Reservado.

/etc/group

Este archivo guarda la relación de los grupos a los que pertenecen los usuarios del sistema, contiene una línea para cada usuario con tres o cuatro campos por usuario:

root:x:0:root
ana:x:501:
sergio:x:502:ventas,supervisores,produccion
cristina:x:503:ventas,sergio
El campo 1 indica el usuario.
El campo 2 'x' indica la contraseña del grupo, que no existe, si hubiera se mostraría un 'hash' encriptado.
El campo 3 es el Group ID (GID) o identificación del grupo.

El campo 4 es opcional e indica la lista de grupos a los que pertenece el usuario

Actualmente al crear al usuario con useradd se crea también automáticamente su grupo principal de trabajo GID, con el mismo nombre del usuario. Es decir, si se añade el usuario 'sergio' también se crea el /etc/group el grupo 'sergio'. Aun así, existen comandos de administración de grupos que se explicarán más adelante.

/etc/login.defs

En el archivo de configuración /etc/login.defs están definidas las variables que controlan los aspectos de la creación de usuarios y de los campos de shadow usadas por defecto. Algunos de los aspectos que controlan estas variables son:

Número máximo de días que una contraseña es válida PASS_MAX_DAYS
El número mínimo de caracteres en la contraseña PASS_MIN_LEN
Valor mínimo para usuarios normales cuando se usa useradd UID_MIN
El valor umask por defecto UMASK
Si el comando useradd debe crear el directorio home por defecto CREATE_HOME

Basta con leer este archivo para conocer el resto de las variables que son autodescriptivas y ajustarlas al gusto. Recúerdese que se usaran principalmente al momento de crear o modificar usuarios con los comandos useradd y usermod.

Archivos de configuración

Los usuarios normales y root en sus directorios de inicio tienen varios archivos que comienzan con "." es decir están ocultos. Varían mucho dependiendo de la distribución de Linux que se tenga, pero seguramente se encontrarán los siguientes o similares:

```
ls -la
drwx------ 2 ana  ana  4096 jul  9 09:54 .
drwxr-xr-x 7 root root 4096 jul  9 09:54 ..
-rw-r--r-- 1 ana  ana    24 jul  9 09:54 .bash_logout
-rw-r--r-- 1 ana  ana   191 jul  9 09:54 .bash_profile
-rw-r--r-- 1 ana  ana   124 jul  9 09:54 .bashrc
```

.bash_profile aquí podremos indicar alias, variables, configuración del entorno, etc. que deseamos iniciar al principio de la sesión.

.bash_logout aquí podremos indicar acciones, programas, scripts, etc., que deseemos ejecutar al salirnos de la sesión.

.bashrc es igual que .bash_profile, se ejecuta al principio de la sesión, tradicionalmente en este archivo se indican los programas o scripts a ejecutar, a diferencia de .bash_profile que configura el entorno.

Lo anterior aplica para terminales de texto 100%.

Si deseamos configurar archivos de inicio o de salida de la sesión gráfica entonces, en este caso, hay que buscar en el menú del ambiente gráfico algún programa gráfico que permita manipular que programas se deben arrancar al iniciar la sesión en modo gráfico.

En la mayoría de las distribuciones existe un programa llamado "sesiones" o "sessions", generalmente está ubicado dentro del menú de preferencias. En este programa es posible establecer programas o scripts que arranquen junto con el ambiente gráfico, sería equivalente a manipular 'bashrc'.

Además Linux permite que el usuario decida qué tipo de entorno Xwindow a utilizar, ya sea algún entorno de escritorio como KDE o Gnome o algún manejador de ventanas como Xfce o Twm.

Dentro del Home del usuario, se creará un directorio o archivo oculto "." , por ejemplo '.gnome' o '.kde' donde vendrá la configuración personalizada del usuario para ese entorno. Dentro de este directorio suele haber varios directorios y archivos de configuración.

Estos son sumamente variados dependiendo de la distribución y del entorno. No es recomendable modificar manualmente (aunque es perfectamente posible) estos archivos, es mucho más sencillo modificar vía las interfases gráficas que permiten cambiar el fondo, protector de pantalla, estilos de ventanas, tamaños de letras, etc.

A continuación te presento un resumen de los comandos y archivos vistos en este tutorial más otros que un poco de investigación:

Comandos de administración y control de usuarios	
adduser	Ver useradd
chage	Permite cambiar o establecer parámetros de las fechas de control de la contraseña.
chpasswd	Actualiza o establece contraseñas en modo batch, múltiples usuarios a la vez. (se usa junto con newusers)
id	Muestra la identidad del usuario (UID) y los grupos a los que pertence.
gpasswd	Administra las contraseñas de grupos (/etc/group y /etc/gshadow).
groupadd	Añade grupos al sistema (/etc/group).
groupdel	Elimina grupos del sistema.
groupmod	Modifica grupos del sistema.
groups	Muestra los grupos a los que pertence el usuario.
newusers	Actualiza o crea usuarios en modo batch, múltiples usuarios a la vez. (se usa junto chpasswd)
pwconv	Establece la protección shadow (/etc/shadow) al archivo /etc/passwd.
pwunconv	Elimina la protección shadow (/etc/shadow) al archivo /etc/passwd.
useradd	Añade usuarios al sistema (/etc/passwd).
userdel	Elimina usuarios del sistema.
usermod	Modifica usuarios.

Archivos de administración y control de usuarios	
.bash_logout	Se ejecuta cuando el usuario abandona la sesión.
.bash_profile	Se ejecuta cuando el usuario inicia la sesión.
.bashrc	Se ejecuta cuando el usuario inicia la sesión.
/etc/group	Usuarios y sus grupos.
/etc/gshadow	Contraseñas encriptadas de los grupos.
/etc/login.defs	Variables que controlan los aspectos de la creación de usuarios.
/etc/passwd	Usuarios del sistema.
/etc/shadow	Contraseñas encriptadas y control de fechas de usuarios del sistema.

7 IDENTIDADES DE FICHERO

En Linux, los ficheros o directorios tienen 3 tipos de identidades y por tanto tres perfiles.

Perfil propietario
Perfil grupo o grupo asociado
Perfil sistema

Permisos de ficheros y directorios

A cada uno de estos perfiles se les puede asociar unas normas de utilización que son conocidas como los permisos.

Los permisos tienen tres elementos lectura, escritura y ejecución, de tal forma que con la combinación de perfiles y permisos se puede establecer un sistema de administración y control de ficheros y directorios muy eficiente.

En el caso de los ficheros:

- Permiso de lectura (read)
  Si tienes permiso de lectura de un archivo, puedes ver su contenido.

- Permiso de escritura (write)
  Si tienes permiso de escritura de un archivo, puedes modificar el archivo. Puedes agregar, sobrescribir o borrar su contenido.

- Permiso de ejecución (execute)
  Si el archivo tiene permiso de ejecución, entonces puedes decirle al sistema operativo que lo ejecute como si fuera un programa. Si es un programa llamado “foo” lo podremos ejecutar como cualquier comando. Un script (interprete) que necesita permiso de lectura y ejecución, un programa compilado solo necesita ser lectura.

En el caso de los directorios:

- Permiso de lectura en un directorio.
  Si un directorio tiene permiso de lectura, puedes ver los archivos que este contiene. Puedes usar un ls para ver su contenido, que tengas permiso de lectura en un directorio no quiere decir que puedas leer el contenido de sus archivos si no tienes permiso de lectura en esos.

- Permiso de escritura en un directorio.

Con el permiso de escritura puedes agregar, remover o mover archivos al directorio

- Permiso de ejecución en un directorio.
  Ejecución te permite usar el nombre del directorio cuando estas accediendo a archivos en ese directorio, es decir este permiso lo hace que se tome en cuenta en búsquedas realizadas por un programa, por ejemplo, un directorio sin permiso de ejecución no sería revisado por el comando find

Los permisos de lectura, escritura y ejecución se representan con r, w y x respectivamente.

Veámoslo de manera práctica. Cuando en un directorio tecleamos
ls –l

obtenemos para cada directorio o archivo una primera columna con una serie de caracteres relacionados con los permisos:

-rw-rw-r--

El primer carácter representa el tipo de elemento y como se interpreta en Linux

d	directorio
l	enlace simbólico
c	dispositivo especial de caracteres
p	canal
s	socket
-	fichero estándar

Después tenemos un primer bloque de 3 caracteres. Este primer grupo contiene los permisos asociados al propietario del fichero.

El siguiente bloque de 3 caracteres contiene los permisos asociados al grupo y el último y tercer bloque de 3 caracteres los permisos asociados al sistema (por tanto a cualquier usuario).

Cada bloque tiene 3 caracteres que determinan la ausencia o presencia del permiso con la secuencia

rwx todos los permisos

cuando no tenga alguno de los permisos o ninguno de ellos se sustituye cada uno de ellos por guion medio, por ejemplo:

r-- solo permiso de escritura

--- ningún permiso

Para cambiar los permisos empleamos el comando chmod (change mode), que pueden agregar o remover permisos a uno o más archivos con + (mas), – (menos) e = (igual).

Por ejemplo,

chmod –w log1.txt

le quitará el permiso de escritura a ese archivo. Y,

chmod +x log1.txt

Si empleamos el operador = establecerá solamente el permiso que incluyamos, por ejemplo:

chmod =r log1.txt Nota = lo cambia en todos los perfiles
En los casos anteriores solo modifica los permisos de propietario con + y -, si queremos modificar los de grupo o los de sistema, emplearemos las opciones u (usuario), g (grupo) y o (otros).

Por ejemplo:

chmod g+w log1.txt

También podemos emplear con chmod el modo octal.

Como resultado de la combinación de los tres tipos de permisos (lectura, escritura y ejecución), con las tres clases de usuarios (propietario, grupo y otros), se obtiene 2^3= permisos en total que pueden ser asignados o denegados de forma independiente.

La base 8 se utiliza habitualmente para que exista un dígito por cada combinación de permisos (un bit a modo de bandera por cada permiso, con valor 1 ó 0 según el permiso esté concedido o denegado).

Así, las posibles combinaciones se resumen en números octales de tres dígitos del 000 al 777, cada uno de los cuales permite establecer un tipo de permiso distinto a cada clase de usuario.

El primer dígito establece el tipo de permiso deseado al dueño; el segundo al grupo; y el tercero al resto de los usuarios.

Número	Binario	Lectura (*r*)	Escritura (*w*)	Ejecución (*x*)
0	000	✗	✗	✗
1	001	✗	✗	✓
2	010	✗	✓	✗
3	011	✗	✓	✓
4	100	✓	✗	✗
5	101	✓	✗	✓
6	110	✓	✓	✗
7	111	✓	✓	✓

De tal manera que podemos establecer los permisos con chmod seguido de la secuencia de números para cada perfil, por ejemplo:

```
x-----------------------x----------x
|chmod u=rwx,g=rwx,o=rx  | chmod 775 |
|chmod u=rwx,g=rx,o=     | chmod 760 |
|chmod u=rw,g=r,o=r      | chmod 644 |
|chmod u=rw,g=r,o=       | chmod 640 |
|chmod u=rw,go=          | chmod 600 |
|chmod u=rwx,go=         | chmod 700 |
x-----------------------x----------x
```

Cambio de propietario de fichero

Como lo que determina el acceso a un fichero o directorio en Linux son sus permisos respecto a su propietario, grupo de propietario o sistema en general (cualquier usuario), puede ser necesario cambiar los propietarios de un archivo o directorio.

Para ello empleamos:

chown opciones nuevopropietario:nuevogrupo fichero

Por ejemplo

sudo chown lucia:usuarios log1.txt

O de manera recursiva,

sudo chown –R lucia pelis

8 PROCESOS Y TRABAJOS

Salvo los comandos internos, cualquier tarea crea en Linux un proceso identificado con un PID.

Visualización de procesos

Para ver los procesos tenemos el comando

ps

Cuyos resultados se estructuran en un listado con las siguientes columnas:

PID	Identificador del proceso
TTY	N° de la terminal
TIME	Tiempo
Command	Comando

El comando ps dispone de decenas de opciones, por ejemplo:

ps aux muestra todos los procesos en detalle del sistema

ps axjf que mostrará un árbol jerárquico con la ruta del programa al que pertenece el proceso

ps -u usuario para ver los procesos de un usuario

Para ser más precisos se puede buscar un determinado proceso con el que queramos dar con la ayuda de un pipe y el filtro grep

Por ejemplo, para buscar el proceso SSH usamos:

ps aux | grep ssh

Los procesos pueden ejecutar otros procesos (procesos hijos). Los procesos hijos contienen su propio PID pero a su vez también contienen otro campo llamado PPID (Parent Process Identificator) que es el proceso padre al que pertenece el proceso.

Todos los procesos en Linux tienen su proceso padre excepto el proceso 0 que es el que ejecuta el inicio del sistema (PID 1)

Para ver el PPID de los distintos procesos, utilizamos

ps –ef

Para ver los procesos en forma de árbol podemos emplear:

pstree

Y para ver el PID de un proceso por su nombre empleamos:

pidof nombre

Existen también otros programas para ver los procesos como top o htop.

Estados de un proceso

Los procesos durante su ejecución pasan por distintos estados, algunos de ellos:

- S (sleeping) el proceso está en espera.
- R (running) el proceso está en ejecución.
- T (stop) el proceso se encuentra parado.
- D proceso que se encuentra bloqueado a la espera de un recurso.
- Z (zombie) es un proceso que se encuentra en estado zombie, es decir, que es un proceso que ha finalizado pero que su proceso padre sigue en ejecución y no se ha "dado cuenta" de la circunstancia de su proceso hijo.

Finalización de procesos

A los procesos se les manda señales para modificar su comportamiento a través del núcleo.

Esto se realiza mediante el comando kill con sus correspondientes señales e indicando el número de PID. Algunas de las señales que se pueden mandar a los procesos son:

- SIGINT ó – 2, interrumpe un proceso, equivale a pulsar CTRL+C
- SIGKILL ó – 9, mata un proceso y no hay vuelta atrás.
- SIGTERM ó – 15, es como SIGINT pero de una forma más "ordenada". Es la señal que se manda por defecto si al comando kill no se le indicar ninguna señal.
- SIGCOUNT ó – 18, reanudar un proceso que se ha parado por ejemplo con SIGSTOP.
- SIGSTOP – 19, parar un proceso, es igual que pulsar CTRL+Z

La sintaxis sería por ejemplo:

kill -9 4900
kill -SIGKILL 4900

Donde 4900 es el PID del proceso

Con el comando killall también se mandan señales, pero actúa sobre el nombre del programa, y por tanto sobre todos sus procesos. Por ejemplo:

killall Firefox

ó

killall -9 sshd

9 INSTALACIÓN DE PROGRAMAS

Podemos instalar software de manera gráfica en Ubuntu con el Centro de Software, pero el verdadero potencial de las distro Linux a la hora de instalar programas es mediante la terminal.

Para instalar emplearemos la herramienta o programa apt con sus diferentes opciones.

Búsqueda de software

Empleamos apt-cache con las siguientes opciones, por ejemplo:

apt-cache search libreoffice

Que nos devolverá todos los paquetes que incluyan ese nombre con su descripción.

Ó:

apt-cache show libreoffice-10ln-es

que nos muestra un paquete concreto.

Nota: con la opción showpkg muestra más información.

Instalación de software

En primer lugar decir que Ubuntu tiene un listado de paquetes en apt que posteriormente es utilizado para las instalaciones. Para tener ese listado actualizado emplearemos:

apt-get update

Por otra parte, para realizar una actualización segura del sistema (y por tanto de todos los paquetes) empleamos:

apt-get upgrade

Si necesitamos una actualización total del sistema empleamos:

apt-get dist-upgrade

Y definitivamente para instalar paquetes, empleamos apt-get con la opción install, por ejemplo:

apt-get install

Eliminación de paquetes

Emplearemos:

apt-get remove

Si queremos eliminar sus ficheros de configuración añadimos la opción

apt-get remove --purge