

Iniciación básica a GNU/Linux

En este documento voy a explicar brevemente las cosas esenciales que hay que saber para utilizar un sistema operativo con núcleo Linux y aplicaciones GNU.

Visitar:

<http://es.wikipedia.org/wiki/GNU>

<http://es.wikipedia.org/wiki/Linux>

En concreto en este caso me voy a centrar sobre las distribuciones basadas en Debian:

1. Comandos básicos, lenguaje shell:

Hoy en día las versiones de sistemas operativos que utilizan el núcleo Linux permiten realizar cualquier tarea de forma gráfica gracias al servidor gráfico denominado X.

Visitar:

http://es.wikipedia.org/wiki/X.Org_Server

Por esta razón no debemos descartar la utilización de la terminal como herramienta de trabajo, que nos permite realizar desde las tareas más básicas hasta las tareas de prueba, mantenimiento y desarrollo de aplicaciones.

La Terminal es una ventana donde podemos escribir texto. Este texto al pulsar intro es interpretado por un intérprete de comandos.

Visitar:

<http://es.wikipedia.org/wiki/Bash>

El intérprete de comandos normalmente lo que hace es comprobar si el texto que hemos introducido corresponde con el nombre de algún binario que hay en las carpetas: /bin, /usr/bin, /usr/local/bin y otras carpetas que se encuentran en el PATH.

Visitar:

[http://es.wikipedia.org/wiki/Ruta_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Ruta_(inform%C3%A1tica))

Si el interprete de comandos encuentra algún binario con el nombre del comando que hemos introducido, entonces lo lee y ejecuta aplicando las opciones de uso con que hayamos decidido ejecutar ese comando.

Si no encuentra ningún binario con el nombre del comando que hemos introducido entonces indica que no se encuentra el comando. Si hemos especificado alguna opción de ejecución del comando de forma errónea entonces indica que no se puede ejecutar el comando de esa forma y probablemente (dependiendo de la configuración de instalación de dicho comando/binario/programa) nos enseñe las modalidades de uso.

A continuación nos centraremos en algunos comandos básicos que se pueden utilizar con una terminal:

`man comando` :nos enseña el manual del comando en cuestión

`cd` :sirve para acceder a la dirección que le especificamos.

Ejemplos:

`cd ~` :accedemos a la carpeta personal del usuario actual

`cd -` :accedemos a la última carpeta visitada

`cd /` :accedemos al directorio raíz

`mkdir nombredirectorio` :crea un directorio con el nombre indicado

`ls` :imprime por pantalla el contenido de un directorio

`mv` :sirve para mover carpetas u archivos, primero copia el fichero/carpeta de origen a su destino y luego borra el origen

`cp` :copia el fichero/carpeta de una dirección indicada a otra dirección que será el destino preservando tanto fichero/carpeta el origen como destino

`ln` :crea enlaces que pueden ser duros o simbólicos (añadiendo la opción '-s' se crean enlaces simbólicos) a archivos u directorios.

Los enlaces se utilizan para tener acceso al mismo archivo u directorio desde varias direcciones. Los enlaces no son copias de los archivos o directorios, una modificación en un enlace afecta al archivo original. Los enlaces duros ocupan más capacidad en el disco duro del ordenador que los simbólicos. Los enlaces duros permiten restringir la lectura/escritura para que solo puedan escribir en ellos algunos usuarios. Si poseemos un ordenador personal lo más conveniente es utilizar enlaces simbólicos siempre que se necesiten.

`cat fichero` :muestra por pantalla el contenido de un fichero

`echo texto` :muestra por pantalla el texto que sigue al comando `echo`

`rm archivo` :borra el archivo que le sigue al comando 'rm'. Tambien se pueden borrar una carpeta añadiendo la opción '-r'.

`pwd` :imprime por pantalla la dirección del directorio donde esta trabajando momentaneamente la consola

`find /direccion/ -name nombre` :encuentra 'nombre' en la dirección especificada

`tar -cvf archivo.tar archivo` :empaqueta la carpeta o el fichero de nombre 'archivo' en un fichero con la extensión '.tar'

`gzip` :comprime una carpeta o directorio que preceden al comando 'gzip'.

`sudo -s` :Adquirir privilegios de administrador. Todos los comandos que introduzcamos a continuación se ejecutarán con privilegios de administrador. Notese que para utilizar este comando nuestro usuario debe tener privilegios de administrador y debemos conocer la contraseña del usuario que ejecuta el comando (la contraseña de nuestro usuario).

Los comandos básicos se pueden concatenar para formar comandos que se adapten más a las necesidades del usuario. Ejemplo:

```
cp -uv $(find /home/$(whoami)/ -name *.txt | grep *a*) /home/$(whoami)/Documentos
```

El comando antes indicado busca todos los ficheros que se encuentra en nuestra carpeta personal que tengan extension '.txt' y que su nombre contenga la letra 'a' y los copia en la carpeta que se llama Documentos de nuestra carpeta personal.

Al igual que en windows, tambien en Linux se pueden utilizar los comandos básicos para realizar scripts ejecutables (en shell "shellscripsts"). A diferencia de windows en donde los scripts son ficheros '.bat' en linux son ficheros '.sh'

Ejemplo de fichero '.sh':

```
#!/bin/bash
sleep 4
echo "Hola Mundo">>/tmp/hola.txt
```

La linea '#!/bin/bash' especifica que para la ejecución de este script se debe utilizar el interprete de comandos 'bash'. Y a continuación el script lo que hace es esperar 4 segundos y despues escribir 'Hola Mundo' en el fichero 'hola.txt' dentro de la carpeta temporal /tmp.

2. Instalar aplicaciones con la ayuda de la terminal:

En entornos Debian se pueden utilizar los gestores de paquetes apt-get y aptitude

para instalar aplicaciones. Ejemplo:

```
apt-get install aplicación
```

```
aptitude install aplicación
```

Basicamente ambos comandos realizan la misma tarea.

Muy raras veces nos podemos encontrar con que la aplicación que queramos utilizar no se encuentra disponible en los repositorios de la distribución. En ese caso hay varias opciones.

2.1 Compilar código fuente.

Para realizar la compilación de código fuente necesitamos las herramientas de compilación, que se pueden instalar ejecutando:

```
sudo aptitude install build-essential linux-headers-$(uname -r)
```

Este comando nos instalará tanto las herramientas de compilación generales como 'gcc' y 'g++' como las herramientas de compilación especiales de nuestro núcleo Linux en particular.

Es importante investigar si la aplicación que deseamos compilar tiene algún requisito especial. Para la creación de algunos binarios es necesario tener las librerías de desarrollo de esa aplicación. Sin estas lo más probable es que no se compile correctamente la aplicación. No puedo concretar más porque cada aplicación es diferente y cada una tiene sus propios requisitos.

También se debe comprender que compilar no significa instalar. Dentro de entornos Microsoft instalar significa descomprimir y copiar el binario en la carpeta requerida para funcionar. Esto se hace normalmente con la interfaz gráfica de instalación así que el usuario no sabe exactamente lo que sucede.

Mientras que compilar significa crear a partir de código fuente un binario que se utilizará en determinado sistema. Con esto estamos literalmente creando la aplicación en nuestro pc y esta solo funcionará para nuestro pc.

2.1.1 Ficheros '.tar.gz'

Para descomprimir un fichero .tar.gz podemos utilizar el comando:

```
tar -xvzf fichero.tar.gz
```

El fichero que se vaya a descomprimir es preferible que lo hayamos dejado en una carpeta nueva vacía ya que al descomprimirlo no sabemos si contiene una carpeta o archivos sueltos.

Para compilar algunos paquetes vienen con sus propias instrucciones aunque generalmente suelen ser:

```
./configure  
make  
make install
```

'./configure' se utiliza para comprobar de que herramientas de compilación disponemos y cuales serán las opciones al realizar la compilación.

'make' crea los binarios de la aplicación a partir del código fuente.

'make install' se utiliza para integrar los componentes de nuestra aplicación en el sistema operativo. Notese que para 'make install' hay que utilizar privilegios de administrador porque se escriben módulos para el núcleo en sí que irán en carpetas ajenas a la carpeta personal.

2.1.2 Ficheros '.tar.bz2'

Para descomprimir ficheros .tar.bz2 se utiliza el comando:

```
tar -xvjf fichero.tar.bz2
```

Leer:

```
man tar
```

2.2 Aplicaciones ya preparadas para utilizar comprimidas.

Hay algunos casos en los que las aplicaciones que nos bajamos de internet ya son listas para utilizar y solo necesitamos descomprimirlas. En este caso para integrar la aplicación con nuestro sistema operativo simplemente los archivos binarios los tenemos que enlazar a una carpeta que se encuentra en el PATH, como por ejemplo "/usr/local/bin".

```
cd /usr/local/bin  
ln -s /direccion_de_la_carpeta_de_la_aplicacion/binario
```

Normalmente la carpeta de la aplicación contiene 2 carpetas llamadas 'bin' e 'lib' que contienen los binarios de la aplicación y las librerías de la misma.

2.3 Ficheros '.jar'.

Esta extensión en principio es un formato que se utiliza para archivar ficheros parecido a zip pero que incorpora comandos "java". Se instalan de la siguiente forma:

```
java -jar fichero.jar
```

Visitar:

<http://es.wikipedia.org/wiki/Jar>

2.4 Ficheros '.bin'.

Los fichero '.bin' son ficheros ejecutables binarios, que contienen la aplicación y normalmente extrae los ficheros de dicha aplicación en nuestro sistema. Atención, muchas veces al ejecutar un fichero binario no se termina de instalar del todo la aplicación en nuestro sistema sino que se necesitan algunos pasos adicionales como los mencionados anteriormente.

Para ejecutar un archivo '.bin' se utilizan las shell que son los interpretes de comandos. Para saber que interprete de comandos es conveniente ejecutar basta con poner './' delante del nombre del fichero binario que se va a ejecutar de la siguiente forma:

```
./fichero.bin
```

2.5 Ficheros '.deb'

Se instalan ejecutando:

```
sudo dpkg -i fichero.deb
```

o

```
sudo gdebi fichero.deb
```

Si tenemos la aplicación gdebi instalada en nuestro sistema (Ubuntu en general la tiene por defecto) basta con hacer doble click sobre el paquete deb. (De lo contrario para instalar esta aplicación ejecutar "sudo aptitude install gdebi")

2.6 Ficheros '.rpm'

Para instalar ficheros '.rpm' es necesario utilizar la aplicación 'alien'(en entorno Debian).

```
sudo aptitude install alien
```

```
alien -d fichero.rpm
```

Convierte fichero '.rpm' a '.deb'.

```
sudo alien -i fichero.rpm
```

Instala el fichero.rpm directamente sin convertirlo a '.deb'.

3. Instalar aplicaciones '.exe'.

Para utilizar aplicaciones nativas del sistema de Microsoft. Necesitamos instalar en nuestro ordenador el interprete de binarios wine.

`aptitude install wine`

Visitar:

<http://www.winehq.org/>

Wine se puede utilizar tanto en entorno gráfico como a través de la consola. Es recomendable al realizar la instalación de un programa utilizar wine a través de consola por varias razones que se exponen más abajo.

Antes de utilizar wine tenemos que configurar una cuantas opciones sencillas:

`winecfg`

Con este comando se abre la ventana de configuración de wine. Hay 7 pestañas que permiten la configuración de wine. La ventana aplicaciones gestiona las aplicaciones que queremos instalar/desinstalar.

3.1 La pestaña librería nos permite cambiar las librerías '.dll' que utiliza wine por las que le especifiquemos nosotros.

3.2 Gráficos permite seleccionar las opciones gráficas que deseamos utilizar en wine. Seleccionar el soporte de aceleración 3d que aparece con la opción Hardware. Si algunas aplicaciones no se ejecutan correctamente cuando el gestor de ventanas de Linux las controla podemos elegir que se ejecuten en un escritorio virtual.

3.3 Integración de escritorio nos permite asignar carpetas de nuestro sistema operativo Linux real a las carpetas ficticias de Mis Documentos y otras de wine.

3.4 Unidades. La causa primordial de que un programa controlado por wine no funcione es porque no se encuentre en algunas de las particiones asignadas a wine. La idea de partición (dosdrive - disk operating system drive) no se aplica a pie de letra, wine puede asignar una carpeta de nuestro sistema operativo y denominarla por ejemplo D:\ al igual que en el sistema operativo Microsoft.

Para que una aplicación funcione tiene que encontrarse en una carpeta que considere wine como dosdrive por ejemplo en la carpeta:

`/home/usuario/.wine/drive_c/`

A la carpeta drive_c siempre se le asigna la letra C:\ si intentamos cambiar esto wine nos avisará de que no podemos hacerlo.

3.5 La pestaña Audio nos permite elegir el dispositivo audio que pueden utilizar las aplicaciones controladas por wine. Se puede elegir por ejemplo ALSA (Advanced Linux Sound Architecture).

3.6 Acerca de... no nos permite configurar nada, simplemente nos muestra la licencia de wine y su versión.

3.7 Instalación en general:

Si realizamos la instalación desde un cd comprobamos con 'winecfg' que se le ha asignado una letra de dosdrive a la unidad de cd. Si la aplicación la instalamos desde un archivo binario suelto '.exe' copiamos este a la carpeta 'drive_c'. Luego ejecutamos **sin sudo**:

wine binario.exe &

Después de instalar la aplicación, el primer arranque se debe realizar también por consola:

Nos dirigimos al sitio donde tengamos instalada la aplicación:

```
cd /home/usuario/.wine/drive_c/Archivos\ de \ Programas/Aplicación
```

```
wine aplicación.exe
```

En el peor de los casos la aplicación para arrancar necesite una librería propietaria de Microsoft. En este caso nos saltará por la terminal un mensaje que indica 'wine no ha podido encontrar la librería ejemplo.dll necesaria para ejecutar la aplicación'.

En este caso deberemos proporcionar a wine la librería '.dll' necesaria copiándola de la carpeta system32 de cualquier sistema operativo windows a la carpeta /home/usuario/.wine/drive_c/windows/system32. Algunas librerías '.dll' también se encuentran por internet. Aunque hay que tener en cuenta que algunas librerías dll tienen licencia que no permite la distribución sin el acuerdo de Microsoft.

Igualmente que en cualquier sistema operativo Microsoft no se le deben proporcionar a wine cualquier librería '.dll', solo las que se nos pida, ya que proporcionarle librerías que no necesita o sobrescribiendo librerías '.dll' nativas de wine puede hacer que dicha aplicación falle igual que MS Windows.

Después del primer arranque y habernos asegurado de que la aplicación funciona correctamente ya podemos hacer accesos directos a dicha aplicación mediante lanzadores o scripts '.sh' y ejecutar la aplicación mediante un doble click.

4. Virtual Box

<http://www.virtualbox.org/>

Virtual Box es una aplicación multiplataforma que permite gestionar máquinas virtuales. Esto significa que podemos instalar y utilizar 2 sistemas operativos al mismo tiempo.

Esta aplicación permite por lo tanto ejecutar un sistema operativo GNU/Linux desde una sesión MS Windows o MS Windows desde GNU/Linux. Al igual que un ordenador real la aplicación asigna una pequeña partición y un trozo de memoria ram a la máquina virtual donde se instalará el sistema operativo que queremos ejecutar y utilizar.

5. La carpeta 'dev'.

La carpeta 'dev' contiene los archivos que hacen referencia a los componentes físicos del ordenador.

`/dev/sda` o `/dev/hda` representa el disco duro primario del sistema o el lector de cds. Depende de la configuración del sistema.

`/dev/sda1` representa una partición de `/dev/sda`

Ejemplo práctico, si queremos realizar una imagen copia de seguridad de un cd debemos ejecutar con el cd introducido en el ordenador:

```
'cat /dev/hda > /home/usuario/Documentos/imagen_de_cd.iso'
```

`/dev/urandom` sirve para generar bits pseudoaleatorios (aleatorios según un algoritmo).

`/dev/random` genera bits aleatorios a partir del ruido eléctrico ambiental.

`/dev/null` Se puede considerar como toma de tierra del ordenador. Cualquier dato que se escriba en `'/dev/null'` se descarta. Y si intentamos leer de `'/dev/null'` este fichero nos devuelve el valor 'FIN DE FICHERO'.

6. El archivo '/etc/fstab'

Visitar:

<http://es.wikipedia.org/wiki/Fstab>

<http://www.google.es/#hl=es&q=fstab+ntfs-3g>

El archivo 'file system table' o 'fstab' es el que controla las particiones en Linux. El mapa de ficheros de sistema se asemeja a un racimo de uva. Todo cuelga del fichero raíz '/'.

Para editar el fichero 'fstab' se deben utilizar privilegios de administrador (+ un editor de texto) y utilizar los siguientes datos:

<identificador de partición> <punto de montaje> <formato> <opciones> <frecuencia de volcado de seguridad> <comprobación de ficheros de sistema>

El identificador de partición se puede determinar especificando la partición que vamos a utilizar de la carpeta 'dev' por ejemplo:

```
/dev/sda1
```

A veces deseamos montar una memoria extraíble en algún punto determinado del sistema de archivos, por esta razón no podemos confiar en los nombres que el sistema operativo asigna a las memorias extraíbles porque por ejemplo podemos introducir un pendrive en un puerto usb, y se le asigna el nombre '/dev/sdb' y respectivamente '/dev/sdb1' a su partición. Al sacar el pen drive e introducir un disco duro nos podemos encontrar con la no muy grata sorpresa de que al disco duro se le ha asignado el nombre '/dev/sdb' también. Lo cual podría generar conflictos.

En este caso es mejor utilizar para identificar la partición que vamos a utilizar el número de identificación de partición. Este número se puede sacar ejecutando con privilegios de administrador: 'vol_id --uuid <partición>'.
Ejemplo:

```
vol_id --uuid /dev/sda1  
7abe4f35-cf4f-4444-91a3-3401f2f8a354
```

Copiamos el número de identificación de la partición al principio de la línea que vamos a añadir en el fichero fstab, de esta manera:

```
UUID=7abe4f35-cf4f-4444-91a3-3401f2f8a354
```

Luego especificamos el punto de montaje, por ejemplo: '/home/usuario/Documentos'(debe de haber una carpeta en esa dirección con el mismo nombre) y la línea editada debe de quedar más o menos así:

```
UUID=7abe4f35-cf4f-4444-91a3-3401f2f8a354 /home/usuario/Documentos
```

Añadimos el formato de la partición: vfat si es fat32 o fat16, ntfs para NTFS, y el resto de formatos: ext2, ext3, reiserfs etc. Si nuestra partición tiene formato reiserfs así que debe quedar más o menos así:

```
UUID=7abe4f35-cf4f-4444-91a3-3401f2f8a354 /home/usuario/Documentos reiserfs
```

Las opciones que añadimos a nuestra línea en el fichero fstab nos permiten controlar que operaciones se pueden realizar o no en dicha partición:

relatime/noatime - permite la gestión (o no en el caso de noatime) de toda la información relacionada con la modificación de ficheros. Cada vez que accedemos un fichero se guarda la fecha en la que ha sido accedido, (esto se puede evitar con noatime).

auto/noauto - montar automáticamente la partición (o no) al encender el ordenador.

user - cualquier usuario puede leer/escribir (si existe la opción rw también escribir) en la partición

exec - permite la ejecución de programas en dicha partición

rw/ro - permite la escritura en dicha partición (read only es modo solo lectura)

async/sync - especifica si la grabación de datos se hace con la partición montada en modo sincrónico o asíncrono (en modo asíncrono hay un retardo en la grabación de datos, pero la velocidad es notablemente mayor)

umask=000 se debe añadir solo si tratamos con particiones fat32 u ntfs, para tener acceso a ellas como usuario sin privilegios

Después de añadir las opciones nuestra línea quedará así:

```
UUID=7abe4f35-cf4f-4444-91a3-3401f2f8a354 /home/usuario/Documentos reiserfs  
noatime,auto,user,exec,rw,async
```

Ahora debemos especificar si necesitamos un volcado de memoria de emergencia. Básicamente lo que hace esto es guardar todos los errores que ha tenido dicha partición al grabar información. Las opciones son 0 o 1. Nuestra línea queda así:

```
UUID=7abe4f35-cf4f-4444-91a3-3401f2f8a354 /home/usuario/Documentos reiserfs  
noatime,auto,user,exec,rw,async 0
```

Y ahora debemos especificar si el sistema operativo va a comprobar la partición en busca de errores cada cierto número de arranques. También tenemos 2 opciones 0 y 1. Elegimos cualquiera de los 2 valores. Y la línea al final debería quedar más o menos así:

```
UUID=7abe4f35-cf4f-4444-91a3-3401f2f8a354 /home/usuario/Documentos reiserfs  
noatime,auto,user,exec,rw,async 0 0
```

☺ Gracias por leer y ánimo!