

Práctica 5: Instalación de programas en Linux. Distintas posibilidades.

Víctor Villegas Borge

En esta práctica veremos diversas formas que permiten la instalación de aplicaciones o programas en Linux. La mayor parte de las distribuciones (en nuestro caso Linux Mint) trae una serie de programas y aplicaciones predeterminados (como por ejemplo, “OpenOffice” o “nano”), y el propio proveedor de la distribución da facilidades para instalar otro gran número de programas.

Esos no son los únicos programas que podemos instalar en nuestra distribución. Hay programas que desarrollan otros proveedores de software para Linux (como por ejemplo SAGE, el reproductor de Flash o Acrobat Reader) que pueden no formar parte de nuestra distribución (Ubuntu) y cuyo proceso de instalación será necesariamente distinto.

También puede haber programas que no estén disponibles para Linux, pero que podamos instalar dentro de nuestra distribución a través de programas intermediarios, como por ejemplo “Wine”.

Wine es una reimplementación de las librerías (o APIS) Win16 y Win32, propias de sistemas operativos Windows NT (e incluso anteriores). Estas librerías permiten a los programadores de aplicaciones “olvidarse” de los detalles concretos de implementación del sistema operativo y de realizar llamadas al sistema, haciendo que sus aplicaciones interactúen únicamente con Win32 y Win16. Algunas (la mayoría) aplicaciones de usuario en Windows sólo usan llamadas a estas librerías. Wine lo que hace es reimplementar las librerías Win32 y Win16 para el kernel de Linux, permitiendo que las aplicaciones desarrolladas para Windows se puedan ejecutar también sobre el kernel de Linux.



Instalación de aplicaciones:

- A partir del código fuente:

Aunque la mayoría de las aplicaciones disponibles para sistemas GNU/Linux están disponibles en formatos de paquetes preparados para su instalación, hay algunas que tenemos que compilar a partir del código fuente. La compilación depende de nuestro sistema, de la arquitectura de nuestro ordenador, etc., por eso el código fuente de las aplicaciones suele venir acompañado de un programa de configuración que prepara el proceso adaptándolo a cada sistema. A partir de él, realizamos la compilación e instalación (por supuesto, para ello necesitamos tener previamente en nuestro sistema el compilador necesario).

La instalación a partir del código fuente no suele ser la forma más sencilla de instalar un programa. Entre otras cosas, requiere que tengamos en nuestra máquina los compiladores necesarios para el lenguaje de programación en que los códigos fuentes se encuentren (C++, Perl, Java). Dependiendo del tamaño de la aplicación, su proceso de compilación puede llevar incluso varios días.

Sin embargo, sí presenta dos importantes ventajas:

- a) Disponemos en nuestro ordenador de los ficheros fuente del programa instalado. Esto nos permitirá poder explorarlos y tener acceso a las librerías y funciones del programa (en el caso de SAGE, podremos explorar la definición de cada una de sus operaciones matemáticas...).
- b) El proceso de compilación de los ficheros fuente se optimiza para el hardware y SO de que dispongamos, mejorando en algunos casos el rendimiento de la aplicación instalada.

- Archivos binarios:

Son los equivalentes en Linux de los “ejecutables” de Windows. Son ficheros que no hay que descomprimir, desempaquetar, etc.

Simplemente hay que ejecutarlos (para ello habrá que darles los permisos necesarios, recuerda lo que hemos visto en prácticas anteriores sobre permisos de ejecución). Son aplicaciones ya compiladas para una arquitectura específica, que simplemente descomprimos en nuestro ordenador y están directamente preparadas para su uso.

Una de las aplicaciones que podemos instalar así es Filezilla. Visita la página web <http://filezilla-project.org/download.php> y dentro de la misma descarga la versión del programa para plataformas Linux de 64 bits.

La extensión del archivo (tar.bz2) significa que el fichero está comprimido. Por tanto, una vez lo hayas descargado a tu disco duro es necesario que

lo descomprimas. Para llevar a cabo su instalación, abre la terminal y realiza los siguientes pasos:

\$bunzip2 nombre_del_fichero.tar.bz2 (bzip2 es un programa libre que comprime y descomprime ficheros: <http://es.wikipedia.org/wiki/Bzip2>)

\$tar -xvf nombre_del_fichero.tar (tar es un empaquetador de archivos muy usado en Linux: <http://es.wikipedia.org/wiki/Tar>)

Una vez hayas descomprimido y desempaquetado los ficheros de Filezilla, entra en la carpeta que ha aparecido (cd Filezilla3) y luego en la carpeta bin (cd bin); ahí han aparecido varios ficheros ejecutables, uno de los cuales es filezilla. Ejecútalo ahora por medio del siguiente mandato:

```
./filezilla
```

En realidad, has podido comprobar que el proceso de instalación ha quedado reducido a un proceso de descompresión del fichero descargado y de desempaquetado del mismo. Si la aplicación ha sido desarrollada para nuestra plataforma, no tendremos problemas a la hora de hacerla funcionar.

- Paquetes DEB y RPM

Los paquetes son la agrupación instalable de los diferentes archivos necesarios para que un programa funcione.

Los paquetes no son ejecutables, sino que son gestionados por terceras aplicaciones. Además, los paquetes de Linux no traen consigo las librerías compartidas (dependencias) y, llegado el caso de necesitarse, serán instaladas o se nos informará de la necesidad de hacerlo, según el gestor de paquetes que estemos utilizando. Es decir, el programa "A" depende del programa "B", y éste a su vez de "C" (cada uno a su vez puede depender de una o más librerías). Así que, si queremos instalar "A", tendremos que instalar todos los demás.

Esto puede convertirse en una tarea ardua y complicada. Comandos como "rpm" o "dpkg" permiten instalar paquetes como "A" individualmente, pero no resuelven las dependencias. Para hacer esto se crearon los sistemas de instalación de paquetes, lo que permite hacer más simples las cosas.

Las aplicaciones Linux se suministran normalmente en dos formatos o tipos de paquetes, y son dependientes de la distribución:

RPM: se utilizan en distribuciones basadas en Red Hat, Fedora, Mandriva...pueden contener tanto binarios como código fuente.

DEB: distribuciones basadas en Debian (Ubuntu, Linex...). Pueden contener tanto binarios como código fuente.

DEB: Es el formato de paquetes nativo de Debian, y por tanto de Ubuntu. Para instalar estos paquetes sólo tienes que hacer doble click sobre el fichero y automáticamente se lanzará la aplicación gdebi, que se ocupará de instalar el paquete y buscar las dependencias de otros paquetes que pudiera necesitar para su correcta instalación.

Si lo preferimos, también se pueden instalar mediante la línea de comandos, mediante el comando dpkg, que no instala automáticamente las dependencias, sino que se limita a indicarnos durante el proceso de instalación.

RPM: Es un sistema de empaquetamiento que se está convirtiendo en un estándar. RPM proporciona al usuario una serie de facilidades que hacen el mantenimiento del sistema más sencillo de gestionar, ya que mantiene una base de datos de los paquetes instalados y de sus archivos, lo que permite realizar consultas y verificaciones del sistema. Asimismo, otra ventaja es que, al actualizar software, los archivos de configuración se respetan de manera que no sea necesario volver a realizar los ajustes específicos que ya tuviera definidos, o si no fuera posible, realiza una copia de seguridad de los mismos.

- Instalación de paquetes usando gestores de paquetes propios de cada distribución:

Lo primero que cabe destacar es que, en función de la distribución que tengamos instalado existen:

- Ubuntu y derivados:
Existen varias formas de instalar programas en modo texto (desde la consola de comandos), por medio de las aplicaciones/gestores de paquetes aptitude, apt-get y apt.

Estos programas son muy similares, aunque tienen sus diferencias. Para no extender la explicación, se adjunta una página web que explica las principales diferencias y parecidos entre estos programas: <https://juncotic.com/apt-vs-apt-get-vs-aptitude-algunas-notas/>

Algunos comandos que debemos de saber utilizar para instalar paquetes con estos programas son:

Instalar paquetes:

\$sudo aptitude install [nombre del paquete]

\$sudo apt-get install [nombre del paquete]

\$sudo apt install [nombre del paquete]

Desinstalar paquetes:

```
$sudo aptitude remove [nombre del paquete]  
$sudo apt-get remove [nombre del paquete]  
$sudo apt remove [nombre del paquete]
```

Actualizar la lista de paquetes disponibles:

```
$sudo aptitude update  
$sudo apt-get update  
$sudo apt update
```

Actualizar el sistema con las actualizaciones de paquetes disponibles:

```
$sudo aptitude upgrade  
$sudo apt-get upgrade  
$sudo apt upgrade
```

Obtener una lista de opciones del comando:

```
$sudo aptitude help  
$sudo apt help  
$sudo apt-get help
```

- Alpine Linux:

Como ya vimos en la práctica anterior, en Alpine Linux no usamos los programas apt, aptitude o apt-get. De hecho, es normal que en diversas distribuciones se usen diversos programas/comandos para llevar a cabo las instalaciones.

En Alpine se usa el comando:

```
$apk add [nombre del paquete]
```

- Tiendas de aplicaciones (interfaz gráfica):

Algunas distribuciones traen incorporados una “Tienda de aplicaciones”, que viene a ser una interfaz gráfica que nos permite navegar, buscar e instalar diversas aplicaciones. Su uso ya debería serte familiar, por lo que no se entrará en detalle a su explicación. En Linux Mint la “Tienda de aplicaciones” se llama: Gestor de Software.

- Instalar aplicaciones de Windows a través de Wine:

Finalmente, vamos a acabar explicando cómo se pueden instalar aplicaciones de Windows en Linux.

Como fue anticipado previamente, necesitaremos usar WINE si queremos instalar aplicaciones que únicamente han sido diseñadas para ser ejecutadas bajo un S.O. Windows. Esto es útil ya que, si bien es cierto que normalmente existen alternativas en Linux que cubren las necesidades del software que ha sido diseñado únicamente para Windows, es posible que en algún momento nos encontremos con un software muy especializado o que tiene alguna función vital que en Linux no existe.

Para empezar, deberemos instalar Wine en nuestro sistema. En Linux Mint podemos hacerlo a través del comando:

```
$apt install wine
```

Una vez wine esté instalado, podremos usarlo para poder ejecutar aplicaciones/instaladores de aplicaciones diseñadas para Windows.

Por ejemplo, vamos a usarlo para ejecutar un ligero juego que está diseñado únicamente para Windows.

Para ello usaremos el comando:

```
$wine [archivo a ejecutar]
```

Se pide por tanto que el alumno adjunte capturas de pantalla que permitan demostrar que en tu máquina virtual de Linux Mint:

- a) Se ha instalado el programa *Filezilla* a través de la instalación de sus archivos binarios.
- b) Se ha instalado el programa *Google Chrome*. Para llevarlo a cabo, descargar el paquete .deb desde la página oficial de Google.
- c) Se ha instalado el programa *Gimp* a través del gestor de aplicaciones apt, tras haber actualizado la lista de paquetes disponibles
- d) Se ha instalado a través del Gestor de Software la aplicación *Audacity*.
- e) Se ha instalado wine a través de la consola de comandos con el gestor de aplicaciones apt.
- f) Se ha ejecutado el programa windowframe.exe gracias a wine.

Finalmente, busca información sobre los gestores de paquetes: PACMAN y YUM y responde a las siguientes preguntas:

- a) ¿En qué S.O. se utilizan?
- b) ¿Qué comandos requieren usar?