Gestión de software

En GNU/Linux, un paquete se entiende como el conjunto de archivos y herramientas necesarias para la instalación y configuración de aplicaciones en el sistema.

Las aplicaciones, o programas, que podemos instalar en un sistema GNU/Linux pueden ser de 2 tipos: interpretadas o compiladas

Aplicaciones interpretadas

Una aplicación interpretada no es más que un conjunto de instrucciones (comandos) que necesitan de un intérprete para poder ser ejecutada. Este tipo de aplicaciones son más conocidas como programas o scripts. Para su ejecución necesitamos un archivo con las instrucciones (programa) y un intérprete de éstas (shell, Python, ...)

Aplicaciones compiladas

Cuando se trata de una aplicación compilada, disponemos de una serie de archivos (el código fuente) que son escritas en un lenguaje de programación no interpretado. A través de un compilador (C, Pascal, ...) generamos los binarios ejecutables para su ejecución. De esta manera disponemos de un ejecutable propio que no necesita de elementos adicionales (el compilador sólo es necesario para la generación de los archivos binarios - y sus dependencias -).

Los sistemas GNU/Linux en la actualidad permiten en su mayoría de distribuciones 3 opciones de gestión del software que contienen:

- Instalación avanzada desde repositorios
- Instalación básica desde el paquete a instalar
- Compilación de paquetes

Instalación avanzada desde repositorios

Cuando hablamos de instalación “avanzada” desde repositorios queremos hacer referencia a 2 características fundamentales:

1. No disponemos del paquete a instalar
2. Vamos a resolver las posibles dependencias del software a instalar

Para llevar a cabo todo este proceso hemos de diferenciar entre la familia de distribuciones Debian y Red Hat, ya que, si bien el principio de funcionamiento es similar. cada una cuenta con sus propias herramientas.

Los comandos que se utilizan en la instalación avanzada de paquetes son:

- Para Debian el comando APT
- Para RedHat el comando YUM

Repositorio

En GNU/Linux existe el concepto repositorio, que no es más que un gran “almacén” de paquetes candidatos de ser instalados en un SO. De hecho, podemos encontrar repositorios que contengan paquetes para más de una versión de SO, por lo que es muy importante identificar nuestra distribución y su versión (ver /etc/os-release) para configurar correctamente la lista de repositorios a utilizar.

Debian

En los sistemas Debian (y derivados) la lista de repositorios a utilizar por el sistema se guarda en un fichero llamado /etc/apt/sources.list.

Cada una de las líneas de este fichero se compone de 4 campos, que se describen a continuación:

Descripción	**Definición**	
Tipo de paquete	indica si los paquetes disponibles son paquetes instalables (deb) o si se trata de las fuentes del software (deb-src)	
Réplica del repositorio	contiene una URI que identifica el recurso público donde se encuentran los paquetes disponibles	
Versión de SO	especifica qué versión de SO estamos utilizando	
Sección del repositorio	los paquetes disponibles dentro de un repositorio se clasifican en función de su licencia	
	main	sección oficial con todos los paquetes de la distribución ninguno de ellos depende de software fuera de esta sección
	contrib	sección que contiene paquetes que requieren de otros (fuera de la distribución) para funcionar y que cumplen 100% con la DFSG
	non-free	sección que contiene paquetes que requieren de otros (fuera de la distribución) para funcionar y no que cumplen 100% con la DFSG

Por lo tanto, una línea del fichero /etc/apt/sources.list quedaría de la siguiente manera:

```
deb https://download.virtualbox.org/virtualbox/debian buster contrib
```

CentOS

Los sistemas de la familia Red Hat y derivados trabajan de manera un tanto diferente. En este caso los repositorios a utilizar se encuentran dentro del directorio `/etc/yum.repos.d/`, y cada repositorio a utilizar posee de un archivo de configuración.

Dentro del archivo de configuración de cada repositorio podemos encontrar un conjunto de parámetros definidos, siendo los más habituales los que se presentan a continuación:

[repositoryid]	**ID del repositorio**
name=	Nombre identificativo del repositorio
baseurl=	Dirección completa del recurso que contiene los archivos instalables

[repositoryid]	ID del repositorio
mirrorlist=	URL que contiene una lista de “baseurl”
enabled=	0-1, indica si va a ser consultado o no
gpgcheck=	0-1, indica si utiliza o no una clave GPG para firmar los paquetes
gpgkey=	ubicación de la firma GPG que utiliza

Así, un fichero de repositorio de un sistema CentOS sería como el que sigue:

```
[BaseOS]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&
repo=BaseOS&infra=$infra
#baseurl=http://mirror.centos.org/$contentdir/$releasever/BaseOS/$basearch/o
s/
gpgcheck=1
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
```

Instalación básica desde el paquete a instalar

Otro método de instalación de paquetes en sistemas GNU/Linux es utilizando los ficheros de instalación de un programa (paquetes). A diferencia con el método anterior, en este caso SÍ disponemos del paquete que queremos instalar, y NO vamos a resolver sus posibles dependencias de manera automatizada. Como pasaba antes, Debian y Red Hat tienen herramientas distintas para gestionar los paquetes en su distribución:

- Para Debian el comando dpkg
- Para RedHat el comando rpm

Paquete

Un paquete se puede entender como un conjunto de ficheros relacionados entre sí que permiten la instalación de una aplicación. Dentro de este conjunto de ficheros podemos encontrar ejecutables, información de la aplicación o instrucciones para su instalación, todo ello en un único archivo contenedor.

Podemos encontrar diferentes tipos de paquetes en función de los datos que contienen:

binario
: contiene bianrios ejecutables necesarios para la instalación de una aplicación .

fuente
: incluyen el código fuente del programa para poder ser compilado e instalado.

dependencia
: paquete previo necesario para poder instalar el paquete deseado .

virtual
: o *dummy*, consiste en un paquete “vacío” con nombre genérico que provee de otros paquetes .

tarea

agrupaciones de paquetes que iinstalados de forma connjunta proporcionan una funcionalidad única .

Compilación de paquetes

Los desarrolladores de aplicaciones, por norma general, son los encargados de compilar y distribuir los binarios ejecutables de sus programas al público general. En cambio, la comunidad open source, cambia un poco este concepto. De hecho, por definición (basics:creative_commons|creative commons), el código fuente de todos los programas debe estar disponible, por lo que un usuario puede descargar y compilar por sí mismo esa aplicación.

Funcionamiento

El proceso de compilación de paquetes se realiza siguiendo un procedimiento estándar que consta de las siguientes fases

Obtención de las fuentes

El código fuente de una aplicación open source se encuentra siempre disponible, generalmente en el sitio web del proyecto de la aplicación. Por poner un ejemplo, la página https://sourceforge.net se dedica a alojar muchos proyectos de desarrollo open source. La obtención de las fuentes se puede realizar con el comando curl o wget

Extracción de las fuentes

Una vez hemos obtenido las fuentes (por norma general en un paquete comprimido), debemos extraerlas en el directorio donde vayamos a compilar la aplicación. La mayoría de fuentes se encuentran empaquetadas en archivos tar, con la compresión que el desarrolador haya escogido (gzip, bzip2, xz, ...)

Configuración de la aplicación

Para poder compilar una aplicación necesitamos de ciertos requisitos.

- compilador de C
- librerías necesarias
- Archivo de configuración que leerá el compilador

Los 2 primeros requisitos deben estar resueltos antes de iniciar el proceso de configuración. En caso de no ser así, cuando se genere el archivo con las configuraciones (Makefile) el sistema nos devolverá un error.

Para poder resolver la última, los archivos descomprimidos deben contener un comando `configure`, que será el que permita construir el archivo de configuración (Makefile) que el compilador irá leyendo

en su ejecución.

Compilación de la aplicación

Una vez hemos ejecutado el comando `configure` y disponemos del archivo Makefile procedemos a ejecutar el comando `make`. Éste es el encargado de compilar nuestra aplicación y generar los binarios y ejecutables necesarios. Estos archivos se encuentran sólo en el directorio raíz donde hemos descomprimido las fuentes.

Instalación de binarios

El último paso consiste en ejecutar el objetivo `install` del comando `make`. Esto produce la instalación de los binarios compilados (así como de las páginas de manual o archivos de configuración) en las diferentes ubicaciones de destino dentro del FHS.