

Instalación de software al estilo Debian

# A PRUEBA DE TONTOS

Los líos de las dependencias desaparecen con el sistema de paquetes

Debian. **POR BRUCE BYFIELD**



Galyna Andrusheko, 129 RF

**P**uede que haya oído hablar del “dependency hell”, el término utilizado cuando falla la instalación de una aplicación porque necesita una librería de desarrollo que no se encontraba instalada y cuando cada nueva librería que se pretende instalar necesita de decenas de librerías más, reduciendo incluso al linuxero más curtido a un estado de gimoteante frustración.

Afortunadamente, este escenario es en gran parte algo del pasado gracias al sistema de paquetes dpkg y apt-get de Debian. Hace más de una década, los desarrolladores de Debian tuvieron la idea de instalar software dispuesto en paquetes de ficheros que incluían scripts para coger librerías faltantes automáticamente y, de paso, ir configurando sobre la marcha el software. La idea fue copiada finalmente por otros sistemas de paquetes, de modo que ahora nos encontraremos en el infierno de las dependencias sólo cuando intentemos instalar aplicaciones no empaquetadas o que aún están en desarrollo.

El sistema de paquetes Debian se usa en muchas de las distribuciones GNU/Linux más populares. Técnicamente,

dpkg es la herramienta que administra el software, mientras que la mayor parte del tiempo los usuarios se relacionan con dpkg a través de apt-get. Ambas se usan para instalar software desde los repositorios online que se describen en */etc/apt/sources.list*.

También podemos añadir */etc/apt/apt.conf*, el fichero que configura apt-get más detalladamente. Sin embargo, la mayoría de la gente prefiere configurar apt-get a través de subcomandos y opciones. Lo básico de estos comandos es fácil de aprender si deseamos añadir o eliminar software o mantener nuestro sistema. Además, encontraremos que en torno a apt-get ha crecido un subsistema completo de utilidades relacionadas para aumentar nuestro control.

## Añadir y Eliminar

A diferencia de la mayoría de los comandos, apt-get consta de tres partes separadas por espacios: el comando apt-get básico, un subcomando, y el paquete sobre el que se actúa. Por ejemplo, para instalar el juego solitario Pysol escribimos *apt-get* (el comando básico), seguido por *install* (el subcomando) y *pysol* (el nombre del paquete).

Siempre debe haber un subcomando present, aunque para determinadas combinaciones – especialmente para mantenimiento – no se necesita una lista de paquetes. Para incluir múltiples paquetes, o se listan separados por un espacio, o bien se usan expresiones regulares, tales como el asterisco, aunque hay que tener cuidado con esto, ya que puede causar problemas y a veces conducir a resultados imprevistos.

En su lugar, si necesitamos instalar múltiples paquetes, lo más acertado será buscar un metapaquete. Un metapaquete es un paquete “de pega” destinado a simplificar la instalación de aplicaciones grandes que están divididas en más de un paquete. Por ejemplo, en Debian, *kde-minimal* instala el número mínimo de paquetes necesarios para ejecutar el escritorio KDE. Para ver si un metapaquete existe, buscamos online en los repositorios de nuestra distribución; si a pesar de todo falla, adivinamos su nombre y vemos si hemos acertado. Con tal de que utilicemos el sentido común, es poco probable que este método tenga consecuencias irreversibles para nuestro sistema.

El comando básico para añadir o actualizar un paquete de software es

*apt-get install nombre\_de\_paquete*. Tan pronto como introduzcamos el comando, obtendremos un resumen completo de lo que ocurrirá si llevamos a cabo la instalación, incluyendo las dependencias que se instalarán, los paquetes que se actualizarán y los que se eliminarán, y la cantidad de espacio en disco que se necesitará. A menos que la acción pueda continuar automáticamente sin que afecte a nada más, entonces debemos elegir si continuar con el proceso o no (Figura 1). Es recomendable leer el resumen cuidadosamente antes de continuar para asegurarnos de que lo que introdujimos no incluye sorpresas desagradables.

Si usamos un repositorio online no estándar, puede que no esté verificado automáticamente como una fuente válida. Cuando esto ocurre, hemos de continuar sólo si estamos absolutamente seguros de que podemos confiar en el repositorio.

Cuando *apt-get* se pone a funcionar, muestra los paquetes que se están descargando y su progreso, además de la velocidad de descarga y del tiempo necesario para finalizar la operación. Los tiempos son sólo estimados y cambiarán cuando cambie la velocidad de conexión a Internet. Una vez completadas las descargas, *apt-get* instala el software, deteniéndose a veces para realizar preguntas acerca de cómo deseamos que se instale. Una vez que todo esté hecho, si fuera necesario, *apt-get* presenta un resumen de todos los problemas que ha encontrado. Como toque final, el software instalado es añadido a los menús del escritorio.

El comando básico para instalar software puede modificarse con algunas opciones. Por ejemplo, usaremos *-s* para simular la instalación sin hacer nada, asegurándonos así de descubrir todos los problemas antes de realizar la instalación real. Si esta opción nos informa de algún problema, podemos ejecutar el comando de nuevo, esta vez con la opción *-f*, con la esperanza de que *apt-get* pueda proporcionarnos de forma inteligente una solución al problema, o con *-m* para ignorar pérdidas de dependencias, con la esperanza de que obtendremos resultados con los que podamos vivir. Si no deseamos responder a preguntas sobre la instalación, podemos utilizar *-y* para responder a todas con “yes” – una opción peligrosa que es preferible

evitar si no sabemos qué estamos haciendo. Si queremos re-instalar la misma versión de un paquete, usaremos *--reinstall*.

Sin embargo, quizás la opción más útil para la instalación es *-t repository*, la cual permite especificar el repositorio online desde el que deseamos instalar los paquetes y todas sus dependencias. Esta opción es especialmente útil en Debian, cuyos repositorios principales – estable, de pruebas e inestable – describen el estado del software. Por ejemplo, si deseamos la ultimísima versión de Gnome, incluso sin haber sido testeada, podemos introducir *apt-get unstable install gnome-desktop environment*. De modo parecido, en otras distribuciones basadas en Debian podemos añadir una rama de desarrollo del software a nuestros repositorios o una versión de software desarrollada en privado que sólo usaremos de manera ocasional. Con esta opción podemos descargar un paquete cuando la versión más reciente está en fase beta o sin depurar o no funciona.

Como alternativa podremos añadir */repository* al final del comando. Sin embargo, esta opción sólo instalará los paquetes especificados desde la fuente especificada, por lo que puede que no funcione bien.

Si somos un experto, podemos descargar un único paquete a nuestro disco duro para la instalación. En tal caso, usaremos *dpkg* directamente. Por ejemplo, si descargamos una versión de desarrollo del administrador de imágenes *digiKam*, podremos instalarlo cambiando al directorio que contiene el paquete e introduciendo *dpkg -i digikam*.

Para otras opciones, la estructura del comando es la misma, excepto para el cambio en el subcomando. Incluso las opciones disponibles son las mismas, a pesar de que algunas puedan no tener mucho sentido con cada subcomando. El subcomando *remove* desinstala software, mientras que *purge* elimina todas las trazas de él desde nuestro ordenador

```

nanday:~# apt-get install wesnoth
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libboost-regex1.37.0 libboost-iostreams1.37.0
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  wesnoth-all wesnoth-aoi wesnoth-core wesnoth-data wesnoth-dbg
  wesnoth-did wesnoth-editor wesnoth-ei wesnoth-http wesnoth-l wesnoth-low
  wesnoth-music wesnoth-nr wesnoth-sof wesnoth-sothe wesnoth-thot wesnoth-tools
  wesnoth-trow wesnoth-tsg wesnoth-ttb wesnoth-utbs
The following packages will be upgraded:
  ttf-wqy-zenhei wesnoth wesnoth-all wesnoth-aoi wesnoth-core wesnoth-data
  wesnoth-dbg wesnoth-did wesnoth-editor wesnoth-ei wesnoth-http wesnoth-l
  wesnoth-low wesnoth-music wesnoth-nr wesnoth-sof wesnoth-sothe wesnoth-thot
  wesnoth-tools wesnoth-trow wesnoth-tsg wesnoth-ttb wesnoth-utbs
23 upgraded, 0 newly installed, 0 to remove and 1207 not upgraded.
Need to get 255MB of archives.
After this operation, 11.0MB of additional disk space will be used.
Do you want to continue [Y/n]?
    
```

Figura 1: Antes de hacer nada, *apt-get* explica qué hace y nos da la opción de no realizar la operación.

(aunque ninguno de los dos elimina dependencias, por lo que puede que necesitemos ejecutar algunos de los subcomandos de mantenimiento listados más abajo). Si deseamos actualizar cada paquete en nuestro ordenador, entonces usaremos el comando *dist-upgrade* en vez de introducir cada paquete individualmente. Para añadir cada aplicación disponible en los repositorios, esté ya instalada o no, podemos utilizar *upgrade*, aunque esta opción no es útil para la mayoría de la gente.

Casi todo el mundo usa el sistema de paquetes Debian para instalar los ficheros binarios precompilados. Sin embargo, si queremos asegurarnos de que todo nuestro software se ejecuta en nuestro sistema tan eficientemente como sea posible, podemos utilizar el subcomando *source* para descargar los paquetes fuente y la opción *-b* para compilarlos en nuestra máquina. Si los fuente requieren dependencias, podemos hacer uso del subcomando *build-dep* para proporcionarlos. Nótese, sin embargo, que la compilación de los paquetes fuente puede llevar un tiempo considerable, particularmente en aplicaciones grandes – puede que incluso sean horas con una aplicación como OpenOffice.org.

## Mantenimiento del Software

Para ayudarnos con estas operaciones básicas, *dpkg* y *apt-get* incluyen algunas utilidades. Cuando nos topamos con dificultades y estamos buscando información, el comando *dpkg-query* nos la puede dar de forma detallada sobre los paquetes involucrados. Por ejemplo, si escribimos *dpkg-query -p kdepim*, recibimos una descripción del paquete que lista la información de contacto para los desarrolladores que lo mantienen, las

```
nanday:~# dpkg-query -p kdepim
Package: kdepim
Priority: optional
Section: kde
Installed-Size: 68
Maintainer: Debian Qt/KDE Maintainers <debian-qt-kde@lists.debian.org>
Architecture: all
Version: 4:4.3.0-1
Depends: akregator (= 4:4.3.0-1), kaddressbook (= 4:4.3.0-1), kalarm (= 4:4.3.0-1), kdepim-kresources (= 4:4.3.0-1), kdepim-wizards (= 4:4.3.0-1), kmail (= 4:4.3.0-1), knode (= 4:4.3.0-1), knotes (= 4:4.3.0-1), konsolecalendar (= 4:4.3.0-1), kontact (= 4:4.3.0-1), korganizer (= 4:4.3.0-1), ktimetracker (= 4:4.3.0-1), kdepim-strigi-plugins (= 4:4.3.0-1), kjots (= 4:4.3.0-1), kpilot (= 4:4.3.0-1), kleopatra (= 4:4.3.0-1)
Suggests: kdepim-doc
Size: 16052
Description: Personal Information Management apps from the official KDE release
 KDE (the K Desktop Environment) is a powerful Open Source graphical desktop environment for Unix workstations. It combines ease of use, contemporary functionality, and outstanding graphical design with the technological superiority of the Unix operating system.

 This metapackage includes a collection of Personal Information Management (PIM) applications provided with the official release of KDE.
Homepage: http://pim.kde.org/
```

Figura 2: La utilidad `dpkg-query` nos dice todo lo que necesitamos saber sobre cada paquete de software.

dependencias del paquete, tamaño y descripción, además de la página de inicio para el equipo de desarrollo (Figura 2). De modo similar, podemos utilizar la opción `-s` para determinar el estado del fichero o `-L` para ver una lista de todos los ficheros incluidos en el paquete de la aplicación. Toda esta información puede ser de gran valor si nos topamos con problemas, tanto si deseamos resolverlos nosotros mismos como si queremos encontrar a alguien que nos ayude.

El comando `apt-get` incluye algunas otras utilidades en forma de subcomandos que se utilizan sin referencia a ningún paquete. Al igual que usamos `fsck` para investigar y reparar la estructura de un sistema de ficheros, podemos usar `apt-get check` para asegurarnos de que el sistema de paquetes está en perfecto estado.

Cuanto más instalemos y desinstalemos, más debemos ejecutar `apt-get` con los subcomandos `clean` y `auto-clean`. El primero de ellos elimina todos los paquetes que hemos descargado e instalado, mientras que `autoclean` elimina todos los que no pueden descargarse más. Ejecutando ambos ocasionalmente, podemos liberar espacio extra en nuestro disco duro sin que afecte al sistema.

Otro subcomando de mantenimiento útil para `apt-get` es `autoremove`, que elimina paquetes huérfanos (por ejemplo, los que no cumplen ningún propósito porque se añadieron como dependencias para una aplicación que ha sido eliminada con anterioridad). Como estos huérfanos no hacen nada pero ocupan espacio en nuestro disco duro, también es buena idea eliminarlos. A diferencia de `clean` y `autoclean`, el sistema de

paquetes Debian lleva la cuenta de los huérfanos y nos recordará que existen cuando ejecutamos `apt-get` para otro propósito.

Otra opción de mantenimiento es la de añadir o eliminar repositorios de `/etc/apt/sources.list`. La edición de este fichero puede realizarse con `vim`, `emacs` o en cualquier otro

editor de texto. El fichero `sources.list` lista todos los repositorios online que usan `apt-get` y `dpkg`. Cada repositorio se lista en su propia línea conforme a un sistema simple. La entrada para cada uno comienza con `deb` si es un repositorio de binarios y `deb-src` si se trata de uno de paquetes fuente. Esta información va seguida de la URL del repositorio, nombre y subsecciones. Las fuentes en principio se encuentran deshabilitadas con una marca almohadilla (`#`) al comienzo de la línea. Típicamente, estas mismas marcas también son usadas para añadir comentarios que los humanos podemos usar para identificar los fuentes.

Después de añadir y eliminar un repositorio de `sources.list`, debemos ejecutar `apt-get update` para cambiar los repositorios que están usando `apt-get` y `dpkg`. En caso contrario, el sistema de paquetes Debian continúa usando aquellos previamente identificados. La edición tras la actualización tarda unos cuantos minutos para completar cada vez, aunque posee la ventaja de asegurarse de que conocemos con precisión qué fuentes estamos usando. Por esta razón, algunos usuarios prefieren editar `sources.list` y especificar la opción `-t` para establecer desde qué fuentes se instalan. De este modo, las posibilidades de cometer un error son mínimas.

### Utilidades Relacionadas

Dependiendo de nuestra distribución, `apt-get` y `dpkg` pueden tener determinado número de utilidades con las cuales están asociadas. Por ejemplo, la mayoría de las distribuciones probablemente tengan `apt-cdrom`, que puede

especificar una fuente para la instalación desde CD o DVD, aunque esta utilidad suele sólo utilizarse en una primera instalación.

Otra utilidad menos común es `apt-spy`, que puede usarse para determinar el repositorio más rápido para usar. El único problema es que las velocidades de conexión pueden variar dependiendo del número de usuarios, y necesitaríamos ejecutar `apt-spy` varias veces antes de utilizar sus resultados para editar `/etc/apt/sources.list`.

Si somos prudentes, desearemos ver si nuestra distribución incluye `apt-listbugs` antes de instalar un paquete. Esta utilidad busca cualquier error que se ha notificado para la versión del software que estamos a punto de descargar.

Con diferencia, la utilidad de paquetes más útil es `apt-cache`, que ofrece un tesoro de información sobre paquetes y sobre nuestro sistema. Por ejemplo, `apt-cache showpkg nombrepaquete` muestra las versiones que hemos instalado, la última versión disponible y los repositorios que estamos usando, y las dependencias inversas de los paquetes (por ejemplo, qué paquetes dependen de ella).

De manera similar, `apt-cache dump` lista todos los paquetes que hemos instalado, y `apt-cache stats` ofrece información tal como el número de paquetes instalados y el número total de dependencias. Una opción especialmente útil es `apt-cache search nombrepaquete`, que localiza el nombre exacto del paquete o los paquetes que queremos instalar.

### Configuramos el Ejemplo

La combinación `dpkg` y `apt-get` tiene más de doce años de antigüedad. En cierta medida, está comenzando a mostrar su edad, con algunas opciones antiguas que pocos, o ningún usuario moderno utiliza, y con mejoras tendentes a entrar en utilidades relacionadas en vez de en `dpkg` o `apt-get`. El resultado es un juego de herramientas que es a veces pesado y al que lleva su tiempo acostumbrarse.

A pesar de ello, la combinación `dpkg/ apt-get` mantiene la potencia si nos tomamos un tiempo en aprenderla. No es sorprendente que, siguiendo su ejemplo, otros sistemas de paquetes, como `yum` o `urpmi`, hayan racionalizado también sus procesos de instalación. ■