

PAQUETES.

Como ya se expresó anteriormente una distribución de Linux ofrece básicamente un conjunto de paquetes que contienen aplicaciones, utilidades, documentación y el propio kernel del sistema.

Estos paquetes no son más que ficheros con cierto formato que, manipulados por un comando u otra aplicación son capaces de instalarse en el sistema de acuerdo a las especificaciones que determinó su fabricante. Entre las especificaciones puede citarse a los ficheros y directorios que crea el paquete, el tamaño que ocupa una vez instalado, una descripción breve y otra más amplia de su utilidad, las dependencias que puede haber respecto a otros paquetes, etc.

Algunos paquetes son dependientes de plataforma como es el caso del que contiene al kernel, otros por el contrario se pueden instalar en cualquier arquitectura como son los que agrupan documentación. Usualmente los programas fuentes también se agrupan en paquetes independientes de los compilados.

En Linux no se utiliza un único sistema de paquetes, sino que podemos encontrar varias alternativas. Veamos ahora las más usadas:

1. **Paquetes RPM.** Los paquetes RPM (Red Hat Package Manager), deben su nombre a que fueron creados por Red Hat, aunque más adelante fue adoptándose por otras distribuciones como SUSE, Mandriva o Fedora.

Los paquetes se identifican por la siguiente sintaxis:

nombre-versión_aplicación-versión_paquete.arquitectura.rpm

Para instalar paquetes RPM en modo texto se usa el comando rpm.

Por ejemplo si se desea descargar un programa como Virtual Box se podría utilizar el siguiente comando:

```
rpm -i VirtualBox-4.1-4.1.18_78361_fedora17-1.x86_64.rpm
```

2. **Paquetes DEB.** Creados por la distribución Debian y usadas en muchas otras, sobre todo Ubuntu. Están mantenidos por una gran comunidad de desarrolladores y el proyecto está patrocinado por la empresa Canonical Ltd., que ofrece soporte y servicios para Ubuntu.

Estos paquetes se gestionan mediante los comandos apt, aptitude y dpkg. También se puede instalar alguna aplicación gráfica de gestión de paquetes como Synaptics, que permite instalar, desinstalar, actualizar o ver si está instalado un paquete.

3. **Paquetes tgz.** Contienen el código fuente del programa con los archivos necesarios para compilar e instalar el programa mediante los comandos tar, para empaquetar, y gzip o bzip para comprimir. Este tipo de paquetes, propio de la distribución Slackware, se suele usar en casi todas las distribuciones. El paquete está en un tipo de fichero .tar.gz o .tar.bz2.

RPM

Existen varios tipos de formato para definir un paquete. Red Hat introdujo la forma RPM (RedHat Package Manager) la cual ha sido adoptada por otras distribuciones como SuSE y Mandrake. En cambio Debian y Slackware hacen sus paquetes siguiendo otra técnica. Los paquetes RPM por lo general poseen extensión rpm. Estos pueden tener un grado mayor o menor de compatibilidad con las distribuciones que emplean el mismo formato.

En Red Hat la herramienta por excelencia para administrar paquetes es el comando rpm que posee muchísimas opciones. Internamente rpm se basa en una base de datos con información acerca de lo instalado, las dependencias existentes, etc. Esta base de datos es actualizada transparentemente por rpm cuando se instala, actualiza o desinstala un paquete. También puede ser consultada para conocer información diversa acerca de lo ya instalado en el sistema.

De forma general el comando rpm brinda las siguientes posibilidades:

- Instalar, actualizar y desinstalar paquetes.
- Construir y compilar paquetes.
- Inicializar y reconstruir la base de datos.
- Hacer consultas sobre paquetes instalados o no.
- Verificar dependencias.
- Adicionar firmas y chequearlas (mecanismo que permite asegurar la validez e integridad de un paquete).

A continuación se listan las opciones que se emplean más a menudo:

- -i : permite instalar los paquetes. Como argumento se colocan los nombres de los ficheros rpm a instalar. Internamente el comando chequea las dependencias funcionales, y de ser posible, reordena los ficheros rpm para que se satisfagan dichas dependencias durante la instalación.
- -e : permite desinstalar los paquetes. Como argumento se indican los nombres de los paquetes a desinstalar. El nombre de un paquete una vez instalado puede tener dos formas: una breve y otra más amplia que incluye el número de la versión. Algunos paquetes permiten que se instalen simultáneamente distintas versiones del mismo. Tal es el caso del paquete que contiene al kernel cuyo nombre breve siempre es kernel, mientras que pueden haber varios nombres ampliados como kernel-2.4.2-2 ó kernel-2.4.3-12.
- -U : permite actualizar los paquetes de versiones inferiores a superiores. Al igual que en la instalación, se colocan como argumentos los nombres de los ficheros rpm que se desean actualizar. También aquí el comando realiza el ordenamiento más adecuado para satisfacer dependencias. De no existir una versión inferior previamente instalada, el proceso sería equivalente a una instalación de la versión superior.
- -F : es igual a la opción anterior salvo en que hace la actualización sólo si hay una versión inferior del paquete previamente instalada.
- -q : permite hacer consultas diversas a los paquetes instalados. Para ello se emplean como argumento los nombres de los paquetes: de forma breve o no. El tipo de consulta se expresa a través de una segunda opción. Las más importantes son:
- -a : lista los nombres (incluyendo la versión) de todos los paquetes instalados en el sistema.
- -i : brinda información acerca de un paquete: nombre, versión, tamaño, descripción, fabricante, fecha de instalación, licencia, clasificación, etc.
- -f : indica dado un fichero del file system, el nombre del paquete que lo creó.
- -l : lista los nombres de todos los ficheros que instaló un paquete.
- -p : permite hacer consultas a un paquete no instalado a partir del fichero rpm que lo contiene. Se puede combinar con las opciones anteriores -i y -l.

Otras opciones interesantes:

- -v : activa el modo explicativo durante la instalación o actualización. Si se indica dos veces, o sea -vv entonces se imprime información detallada de todas las operaciones efectuadas en la base de datos de RPMs durante el proceso.
- -h : muestra hasta 50 caracteres “#” que expresan el progreso del proceso de instalación o de actualización. Usualmente se combina con -v.
- -R : es una opción de consulta que muestra las dependencias que tiene un paquete.
- -d : es una opción de consulta que lista los ficheros de documentación que instala un paquete.
- -changelog : es una opción de consulta que muestra los logs que expresan los cambios efectuados en el paquete a partir de su versión anterior.
- -scripts : es una opción de consulta que muestra los scripts que se ejecutan al instalar y desinstalar un paquete.

MANIPULACIÓN DE PAQUETES EN DEBIAN.

En el principio existían los .tar.gz. Los usuarios tenían que descomprimir, destaar y compilar cada programa que quisieran usar en su sistema GNU/Linux. Cuando Debian fue creado, se decidió que el sistema incluyera un programa que se encargara de manejar los paquetes (programas) instalados en el ordenador. Este programa se llamó **DPKG**.

Así fue como nació el primer "paquete" en el mundo GNU/Linux, aún antes de que RedHat decidiera crear su propio sistema de paquetes "rpm".

Rápidamente llegó un nuevo dilema a las mentes de los creadores de GNU/Linux. Ellos necesitaban un modo fácil, rápido y eficiente de instalar programas, que manejara automáticamente las dependencias (programas que dependen de otros) y se hiciera cargo de la configuración mientras se actualizan. Nuevamente Debian fue pionera y creó el **APT**, Herramienta Avanzada de Empaquetamiento (Advanced Packaging Tool).

Como parte de su funcionamiento, APT utiliza un archivo que contiene las direcciones de varios servidores (repositorios) que se encuentran en Internet que contienen los paquetes candidatos a ser instalados. También indicamos en este fichero si vamos a cargar paquetes desde medios locales como un cdrom, un directorio compartido de la red, etc. Este archivo es /etc/apt/sources.list. Veamos el contenido de un archivo sources.list:

```
usuario@debian6:~$ cat /etc/apt/sources.list
#
# deb cdrom:[Debian GNU/Linux 6.0.0 _Squeeze_ - Official Multi-architecture amd64/i386 NETINST
#1 20110205-14:45]/ squeeze main
#deb cdrom:[Debian GNU/Linux 6.0.0 _Squeeze_ - Official Multi-architecture amd64/i386 NETINST #
1 20110205-14:45]/ squeeze main
deb http://ftp.es.debian.org/debian/ squeeze main
deb-src http://ftp.es.debian.org/debian/ squeeze main
deb http://security.debian.org/ squeeze/updates main
deb-src http://security.debian.org/ squeeze/updates main
deb http://ftp.es.debian.org/debian/ squeeze-updates main
deb-src http://ftp.es.debian.org/debian/ squeeze-updates main
usuario@debian6:~$
```

Cada línea en este fichero es un repositorio de software, que almacena paquetes que podemos descargar e instalar.

La primera palabra en cada línea, deb o deb-src, indica el tipo del archivo que se almacena en el repositorio: ya sea que contenga paquetes binarios (deb), esto es, los paquetes precompilados que normalmente se usan, o los paquetes fuente (deb-src), que son los códigos originales o fuentes.

A continuación de esta palabra viene la dirección donde se encuentra el repositorio, que puede ser bien una URL de red o un indicador como cdrom.

A continuación, viene el nombre de la distribución de Linux que queremos manejar con el repositorio. Normalmente un repositorio almacena paquetes para varias versiones de Linux, así por ejemplo el fichero anterior indica al repositorio que queremos los paquetes de la versión squeeze (Debian 6.0).

También se indica el tipo de repositorio, que puede ser normal, de actualización, de seguridad, etc.

Cada línea termina indicando el tipo de repositorio que queremos usar. Vemos que en el ejemplo todos nuestros repositorios son main (general).

En Debian normalmente se pueden indicar repositorios para stable, testing, unstable y experimental, que son las distintas versiones de desarrollo ofrecidas por Debian.

En Ubuntu podemos indicar main, restricted, universe y multiverse, dependiendo de si nos queremos bajar los paquetes oficiales y libres de Ubuntu, los paquetes no enteramente libres, paquetes no oficiales, etc.

Siempre que se modifica el archivo sources.list, hay que ejecutar el comando **apt-get update**. Debe hacer esto para permitir a APT obtener la lista de paquetes desde las fuentes que especificamos.

Si queremos utilizar el CD-ROM actual introducido para instalar los paquetes o para actualizar el sistema con APT, lo podemos agregar al archivo sources.list. Para hacerlo, podemos utilizar el programa apt-cdrom así:

```
apt-cdrom add
```

Si tenemos en la unidad de cdrom un cd con paquetes debian, esta instrucción lo montará, y buscará la información de los paquetes en el CD.

El sistema de paquetes utiliza una base de datos para llevar un control sobre los paquetes instalados, los no instalados y cuales están disponibles para su futura instalación. El programa **apt-get** utiliza esta base de datos para averiguar cómo instalar los paquetes que son requeridos por el usuario y para indagar sobre que paquetes adicionales serán requeridos para que el paquete seleccionado funcione correctamente. Esta base de datos se actualiza con la orden **apt-get update**.

INSTALAR PAQUETES

Con el archivo sources.list listo y la lista de paquetes disponibles al día, todo lo que necesitamos es ejecutar apt-get para tener el paquete que queramos instalar. Por ejemplo, si ejecutamos:

```
apt-get install dopewars
```

APT buscará en su base de datos para encontrar la versión más reciente del paquete dopewars y lo descargará del servidor correspondiente especificado en sources.list. Si este paquete necesitara otro para funcionar APT resolverá las dependencias e instalará los paquetes necesarios. Observemos este ejemplo:

```
usuario@debian6:~$ sudo apt-get install 3dchess
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  xaw3dg
Se instalarán los siguientes paquetes NUEVOS:
  3dchess xaw3dg
0 actualizados, 2 se instalarán, 0 para eliminar y 21 no actualizados.
Necesito descargar 195 kB de archivos.
Se utilizarán 607 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
```

Como vemos hemos indicado que queremos instalar el paquete 3dchess, apt se ha puesto en contacto con el repositorio indicado en sources.list, le ha pedido información sobre dicho paquete y ha encontrado que tiene una dependencia no resuelta, es decir, que le hace falta un paquete para funcionar que no tenemos instalado en nuestro sistema (en el caso del ejemplo, el que falta es el paquete xaw3dg).

Vemos como apt automáticamente marca dicho paquete para instalarlo y nos pide confirmación. APT se encargará de bajar ambos paquetes, descomprimirlos, instalarlos en el sistema y configurarlos para que funcionen juntos.

APT sólo pregunta por confirmación cuando se van a instalar paquetes que no fueron especificados en la línea de comando.

Las siguientes opciones de apt-get podrían ser útiles:

- h Ayuda de la orden.
- d Solo descarga los paquetes, no instala nada en el sistema.
- f Continúa aunque haya fallos de integridad en los ficheros bajados.
- y Asume SI a todas las preguntas.
- u Muestra una lista de paquetes modificados.

Pueden seleccionarse varios paquetes para instalar en una sola línea. Los archivos descargados son almacenados en el directorio `/var/cache/apt/archives` para su instalación posterior.

Si en lugar de un paquete queremos descargarnos el código fuente del paquete podemos hacerlo usando el comando **apt-get source paquete**.

REINSTALAR UN PAQUETE

Si queremos reinstalar un paquete, (por qué se haya estropeado, hayamos tocado la configuración y ya no funcione, etc.) podemos usar la opción `--reinstall`:

apt-get install paquete --reinstall

```
usuario@debian6:~$ sudo apt-get install mc --reinstall
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
0 actualizados, 0 se instalarán, 1 reinstalados, 0 para eliminar y 21 no actualizados.
Se necesita descargar 0 B/2173 kB de archivos.
Se utilizarán 0 B de espacio de disco adicional después de esta operación.
(Leyendo la base de datos ... 129719 ficheros o directorios instalados actualmente.)
Preparando para reemplazar mc 3:4.7.0.9-1 (usando .../mc_3%3a4.7.0.9-1_i386.deb) ...
Desempaquetando el reemplazo de mc ...
Procesando disparadores para man-db ...
Procesando disparadores para menu ...
Configurando mc (3:4.7.0.9-1) ...
Procesando disparadores para menu ...
usuario@debian6:~$
```

ELIMINAR UN PAQUETE

Si ya no necesitamos utilizar cierto paquete, podemos eliminarlo de nuestro sistema utilizando APT. Para realizar esta tarea sólo debemos escribir:

apt-get remove paquete

```
usuario@debian6:~$ sudo apt-get remove mc
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los siguientes paquetes se ELIMINARÁN:
  mc
0 actualizados, 0 se instalarán, 1 para eliminar y 21 no actualizados.
Se liberarán 6603 kB después de esta operación.
¿Desea continuar [S/n]? s
(Leyendo la base de datos ... 129718 ficheros o directorios instalados actualmente.)
Desinstalando mc ...
update-alternatives: utilizando /usr/bin/vim.tiny para proveer /usr/bin/view (view) en modo automático.
Procesando disparadores para menu ...
Procesando disparadores para man-db ...
usuario@debian6:~$
```


ACTUALIZAR UN PAQUETE

Las actualizaciones de los paquetes pueden realizarse con tan sólo un comando: **apt-get upgrade**.

Podemos utilizar esta opción para actualizar los paquetes de la distribución actual, o bien para actualizar a una nueva distribución, aunque el comando **apt-get dist-upgrade** es una mejor opción.

Es muy útil utilizar este comando con la opción **-u**. Esta opción muestra la lista completa de paquetes que APT actualizará. Sin ella, se estaría actualizando a ciegas. APT descargará las versiones más recientes de cada paquete y las instalará de la manera más apropiada. Es muy importante ejecutar siempre **apt-get update** antes de probar esto.

```
usuario@debian6:~$ sudo apt-get upgrade -u
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se actualizarán los siguientes paquetes:
  apache2.2-bin bind9-host dnstools gdm3 host iceweasel isc-dhcp-client isc-dhcp-common
  libbind9-60 libdns69 libisc62 libisc60 libisc62 liblwres60 libmodplug1 libmozjs2d
  libnss3-1d libtiff4 tzdata x11-xserver-utils xulrunner-1.9.1
21 actualizados, 0 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 16,0 MB de archivos.
Se utilizarán 98,3 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? █
```

BUSCAR UN PAQUETE

Existen algunas interfaces para el APT que lo hacen más fácil de utilizar. Pero nuestro objetivo aquí es aprender a manejar APT puro. Así que, ¿cómo podríamos saber el nombre de un paquete que queremos instalar?

Tenemos numerosos recursos para realizar esa tarea. Empezaremos con **apt-cache**. Este programa es utilizado por APT para mantener su base de datos. Nosotros sólo veremos un poco de sus aplicaciones. Por ejemplo, supongamos que queremos encontrar un emulador de la nintendo DS, y después bajar algunos juegos. Podríamos realizarlo así:

```
usuario@debian6:~$ apt-cache search nintendo
desmume - Nintendo DS emulator
fceu - FCE Ultra - a nintendo (8-bit) emulator
libgme-dev - Playback library for video game music files - development files
libgme0 - Playback library for video game music files - shared library
gbsplay - A Gameboy sound player
gnome-nds-thumbnailer - Nintendo DS roms thumbnailer for GNOME
kamefu-data - Data files for Kamefu
kamefu - KDE All Machine Emulator Frontend for Unix - binary files
libkamefu-dev - Development headers for Kamefu
libkamefu0 - Libraries for Kamefu
```

Veremos que la orden nos devuelve una cantidad muy grande de paquetes relacionados de alguna u otra manera con nintendo. Para refinar nuestra búsqueda, podríamos utilizar una orden como la siguiente:

```
usuario@debian6:~$ apt-cache search nintendo | grep -i "nintendo ds"
desmume - Nintendo DS emulator
gnome-nds-thumbnailer - Nintendo DS roms thumbnailer for GNOME
usuario@debian6:~$ █
```

Parece que hay un paquete prometedor, el desmume. Para ver información sobre dicho paquete usamos el comando **apt-cache show**

```

usuario@debian6:~$ apt-cache show desmume
Package: desmume
Priority: extra
Section: games
Installed-Size: 5048
Maintainer: Debian Games Team <pkg-games-devel@lists.alioth.debian.org>
Architecture: i386
Version: 0.9.6-1-1
Depends: libasound2 (>= 1.0.18), libc6 (>= 2.3.6-6~), libgcc1 (>= 1:4.1.1), libgl1-mesa-glx
libgl1, libglade2-0 (>= 1:2.6.1), libglu1-mesa (>= 2.14.0), libglu1-mesa | libglu1, libgtk2.0
(>= 2.14.0), libosmesa6 (>= 6.5.2-1) | libgl1-mesa-glide3, libpango1.0-0 (>= 1.14.0), libSDL
debian (>= 1.2.10-1), libstdc++6 (>= 4.4.0), zlib1g (>= 1:1.1.4)
Filename: pool/main/d/desmume/desmume_0.9.6-1-1_i386.deb
Size: 1554672
MD5sum: 5045bf82866ac1a2bd8a7de83fe2b441
SHA1: 96a6291b73aa4bed33e672fc325cf16ff3825247
SHA256: 899292aa8b5fa01519d3728066fd452c4ffb99b44bd51ba0d4fe05f91d8afdb0
Description: Nintendo DS emulator
DeSmuME is a Nintendo DS emulator running homebrew demos and commercial games.

```

Como vemos obtenemos mucha información sobre el paquete, como la descripción, el tamaño, las dependencias, etc. Una vez comprobado que es el paquete que realmente queremos, bastaría con instalarlo con un apt-get install.

MANEJO DE PAQUETES CON DPKG.

Aparte de manejar paquetes con apt, también podemos manejarlos a bajo nivel con los comandos dpkg. Veamos algunos de ellos:

Comando dpkg	Explicación.
dpkg -i paquete.deb	A veces podemos bajar directamente un paquete desde internet, normalmente con la extensión .deb. Para instalar uno de estos paquetes basta con usar dpkg con la opción -i.
dpkg -r paquete	Elimina un paquete (remove).
dpkg -P paquete	Elimina un paquete, y además elimina también todos sus archivos de configuración, temporales, etc.
dpkg -l	Orden importante. Nos muestra un listado de todos los paquetes instalados en nuestro sistema.
dpkg -L paquete	Nos muestra información de un paquete ya instalado en el sistema, indicando que ficheros se instalaron y donde.

Un comando que se suele usar a menudo es dpkg-reconfigure. Este comando nos permite reconfigurar un paquete completo del sistema. Algunos usos de este comando habituales suelen ser:

Comando dpkg-reconfigure	Explicación.
dpkg-reconfigure locales	Configuración del idioma usado en los terminales de Debian.
dpkg-reconfigure xserver-xorg	Configura el sistema gráfico.
dpkg-reconfigure -f noninteractive tzdata	Configura el time zone (la zona horaria).