

Curso Linux Admin

Gestor de Paquetes
aptitude / apt-get

--

Paquetes .deb & .rpm

Temario

Formatos de paquetes en GNU/Linux.....	3
Ejemplos de los paquetes más utilizados.....	3
Paquetes .deb.....	3
Paquetes .rpm.....	4
Limitaciones de RPM.....	4
Gestor de Paquetes “Aptitude”.....	4
Opciones de aptitude.....	5
Patrones de aptitude.....	6
Gestor de Paquetes “yum”.....	7
Acciones	7

Formatos de paquetes en GNU/Linux

Los formatos de paquetes en GNU/Linux son distintos formatos de archivo utilizados por los paquetes de software de varias distribuciones GNU/Linux. Existen las siguientes clases principales de paquetes en GNU/LinuxLinux:

- Paquetes pre-compilados o binarios
- Paquetes de código fuente

Ejemplos de los paquetes más utilizados

- **deb**: paquete originario de la distribución Debian GNU/Linux. Es usado también por distros derivadas de Debian, tales como Ubuntu, y muchas otras.
- **ebuild**: un script en Bash que contiene información sobre cómo obtener, compilar e instalar un paquete en el sistema Portage, pkgcore o Paludis originarios de Gentoo. A partir de estos también se pueden crear paquetes binarios. Las distribuciones basadas en Gentoo, utilizan también el sistema ebuilds, compartiendo total o parcialmente mismo árbol de Portage.
- **RPM**: Red Hat Package Manager, originario de Red Hat, ahora es utilizado por muchas otras distribuciones, como también por openSUSE y Mandriva.
- **tgz** o **tar.gz**: el estándar Tar + gzip, posiblemente con algunos archivos de control extra - utilizado por Slackware y otras distribuciones Linux, o a veces al distribuir de forma muy simplepaquetes hechos a mano.
- **MO**: Utilizado por SLAX. Se abre con el Gestor de módulos de SLAX, y a continuación, se instala al menú de KDE como un fichero de configuración de escritorio. Puede ser colocado en la carpeta de los módulos de SLAX CD para instalañ en el arranque.
- **PUP** and **PET**: usado por el libre y liviano Puppy Linux. Se instala solamente haciendo un click sobre el tipo de paquete. El sistema operativo puede ser instalado en una memoria flash por portabilidad, y traer aplicaciones con él.

Paquetes .deb

Deb es la extensión del formato de paquetes de software de Debian y derivadas (e.j. Ubuntu), y el nombre más usado para dichos paquetes. Como Debian, su nombre proviene de Deborah Murdock, ex-esposa del fundador de la distribución Ian Murdock.

El programa predeterminado para manejar estos paquetes es dpkg, generalmente vía apt/aptitude aunque hay interfaces gráficas como Synaptic,PackageKit, Gdebi o actualmente en Ubuntu Software Center 3.0 que simplifican el trabajo.

Los paquetes deb son archivos ar estándar de Unix que incluyen dos archivos tar en formato gzip, bzip2 o lzma: uno de los cuales alberga la información de control y el otro los datos. Un paquete deb también puede ser convertido a otros formato de paquete y viceversa usando la aplicación alien.

Estos paquetes contienen tres archivos:

- **debian-binary**: número de versión del formato deb. Este es "6.0" para las versiones actuales de Debian.
- **control.tar.gz**: la meta-información del paquete, básicamente lo que se muestra cuando ejecutamos un `apt-cache show nombredepaquete`
- **data.tar**, **data.tar.gz**, **data.tar.bz2** o **data.tar.lzma**: - los archivos que se instalan en nuestro sistema operativo

Paquetes .rpm

RPM Package Manager (o RPM, originalmente llamado Red Hat Package Manager, pero se convirtió en un acrónimo recursivo), es una herramienta de administración de paquetes pensada básicamente para GNU/Linux. Es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas. RPM es el formato de paquete de partida del **Linux Standard Base**.

Originalmente desarrollado por **Red Hat** para **Red Hat Linux**, en la actualidad muchas distribuciones GNU/Linux lo usan, dentro de las cuales las más destacadas son **Fedora Linux**, **Mandriva Linux** y **SuSE Linux**. También se ha portado a otros sistemas operativos.

Limitaciones de RPM

Todo gestor de paquetes presenta una serie de problemas. Por ejemplo se pueden citar:

- No permite gestionar varias versiones de librerías. A veces es necesario para que funcionen a la vez programas antiguos y modernos.
- Los paquetes son monolíticos. Las dependencias de un paquete no se conocen hasta que se abre.
- La base de datos de paquetes puede desincronizarse con el sistema de ficheros
- Necesita un gestor de dependencias externo, como yum

Otros gestores de paquetes como dpkg solucionan estos problemas al menos parcialmente, aunque también necesitan gestionar sus dependencias de forma externa, con gestores como apt

Gestor de Paquetes "Aptitude"

Aptitude es un gestor de paquetes para sistemas Debian GNU/Linux, que proporciona la funcionalidad de **dselect** y **apt-get**, así como de muchos elementos adicionales que no se encuentran en cualquiera de los dos programas. Tiene una interfaz en modo texto, basado en

ncurses, para el manejo del sistema de paquetes de GNU/Linux, que tiene fama de resolver mejor las dependencias que apt y eliminar paquetes inútiles. Entre otras, permite a un usuario:

- Visualizar las distintas listas de paquetes
- Realizar tareas de gestión de paquetes, tales como instalar, actualizar y eliminar paquetes
- Realizar búsquedas de paquetes
- Resolver problemas de dependencias

Éstas acciones pueden ser llevadas a cabo desde una interfaz visual o de la línea de comandos. En este curso nos centraremos en el uso de la consola.

Sintaxis:

```
aptitude [opciones] (paquete)
```

Ejemplo:

```
aptitude install nombre_paquete
```

El ejemplo anterior, instala el paquete llamado *nombre_paquete*.

Recuerden que pueden instalarse más paquetes a la vez, simplemente escribimos sus nombres separandolos por un espacio.

Opciones de aptitude

Aptitude acepta distintas opciones para instalar, reinstalar, borrar, etc. A continuación vamos a enumerarlas:

- aptitude **install** *nombre_paquete*: instala el paquete
- aptitude **remove** *nombre_paquete*: elimina el paquete
- aptitude **purge** *nombre_paquete*: purga el paquete
- aptitude **hold** *nombre_paquete*: bloquea el paquete
- aptitude **unhold** *nombre_paquete*: desbloquea el paquete
- aptitude **keep** *nombre_paquete*: mantiene el paquete
- aptitude **reinstall** *nombre_paquete*: reinstala el paquete
- aptitude **markauto** *nombre_paquete*: marca el paquete como instalado automáticamente
- aptitude **unmarkauto** *nombre_paquete*: marca el paquete como instalado manualmente
- aptitude **update**: actualiza la lista de paquetes disponibles en nuestros repositorios
- aptitude **safe-upgrade**: actualiza los paquetes instalados a su versión más reciente. Puede ocurrir que se instalen nuevos paquetes para resolver dependencias a menos que indiquemos lo contrario (orden `--no-new-installs`), pero no elimina paquetes, sólo actualiza los instalados e instala nuevos para resolver dependencias.

A veces, para poder actualizar un paquete es necesario eliminar otro, en éstos casos aptitude safe-upgrade no es útil y debemos recurrir a aptitude full-upgrade.

- aptitude **full-upgrade**: realiza la actualización de los paquetes que tengamos instalados a su versión más reciente, la supresión y/o la instalación de nuevos paquetes, según sea necesario. Esta opción es menos conservadora que la anterior y, por tanto, tenemos más posibilidades de que realice acciones no deseadas, por tanto es muy recomendable revisar que va a hacer antes de aprobarlo, como contrapartida es capaz de actualizar los paquetes que **aptitude update** no puede mejorar.
- aptitude **keep-all**: Cancela todas las acciones programadas en todos los paquetes, limpia las marcas de instalación, eliminación o actualización.
- aptitude **forget-new**: olvida los paquetes marcados como nuevos (equivalente a pulsar la tecla f en la interfaz de aptitude).
- aptitude **search texto**: busca un paquete que contenga el *texto* en el nombre o descripción
- aptitude **show nombre_paquete**: Muestra información detallada sobre uno o varios paquetes, si contiene una tilde (~) o un signo de interrogación (?), será tratado como un patrón de búsqueda y mostrará todas las palabras clave.
- aptitude **clean**: Elimina todos los archivos .deb descargados previamente (normalmente en */var/cache/apt/archives*).
- aptitude **autoclean**: borra paquetes obsoletos dejando los actualizados

Patrones de aptitude

Si el nombre del paquete contiene una tilde (~) o un signo de interrogación (?), será tratado como un patrón de búsqueda y cualquier paquete que cumpla el patrón podrá ser instalado

Si deseamos instalar una versión específica de un paquete debemos añadir = **version**

Ejemplo:

```
aptitude install gimp = 2.4.6-1
```

No siempre ésta acción instala un paquete, podemos añadir al nombre del paquete una especificación de la acción:

- **aptitude install paquete +**: instala el paquete
- **aptitude install paquete + M**: instala el paquete y lo marca como instalado automáticamente
- **aptitude install paquete --**: elimina el paquete
- **aptitude install paquete _**: purga el paquete
- **aptitude install paquete =**: bloquea el paquete, cancela cualquier actividad de instalación, actualización o supresión, e impide que éste paquete se actualice automáticamente en el futuro.
- **aptitude install paquete ::** mantiene el paquete, cancela cualquier actividad de instalación, actualización o supresión, a diferencia de = no impide que éste paquete se actualice automáticamente en el futuro.
- **aptitude install paquete & M**: marca el paquete como instalado automáticamente
- **aptitude install paquete & m**: marca el paquete como instalado manualmente

Gestor de Paquetes “yum”

Yum es un gestor de paquetes que se puede utilizar para el mantenimiento de los sistemas que utilizan paquetes basados RPM

Sintaxis:

```
yum (acciones) (paquete)
```

Acciones

- **yum install:** Instala paquetes con sus dependencias
 - yum install paquete1 paquete2 ...paqueteN
- **yum update:** Te permite actualizar paquetes junto con sus dependencias
 - yum update
 - yum update paquete1 paquete2 ...paqueteN
 - yum update –exclude=ekiga
- yum **remove:** Te permite eliminar paquetes junto con sus dependencias
 - yum remove paquete1 paquete2 .. paqueteN
- yum **list:** Lista todos los paquetes disponibles y/o los paquetes instalados
 - yum list [*all* / *paquete1*] [*paquete2*]
 - yum list installed
 - yum list available
 - yum list updates
- yum **search:** Nos ofrece un listado de resultados de los paquetes producto de la búsqueda por coincidencia de cadena de caracteres en la descripción, resumen y nombre del paquete
 - yum search paquete1
- yum **info:** Nos ofrece información detallada sobre algún paquete
 - yum info paquete1