Elvira Misfud -k idatzia
Astelehena, 2007(e)ko iraila(r)en 17-(e)an 13:00etan

There are no translations available.

nbsp;

Utilización de paquetes .deb en Edubuntu

1. Introducción

Toda distribución basada en GNU/Linux utiliza algún sistema o mecanismo para la creación y/o manipulación del software que incluye. Es decir, cualquier aplicación integrada en la propia distribución o disponible a través de ella, debe cumplir una serie de requisitos organizativos y estructurales que permitan su gestión de forma ordenada y organizada. Este sistema se llama **Gestor de paquetes** y cada distribución utiliza un gestor de paquetes. Por ejemplo Red-Hat utiliza los paquetes .rpm, Slackware utiliza los paquetes .tar, Debian utiliza los paquetes .deb, etc.

La distribución Edubuntu Feisty Fawn utiliza el sistema de paquetes de Debian (.**deb**) pero sigue una clasificación diferente. Para Edubuntu todo el software disponible está agrupado en cuatro secciones o componentes, dependiendo del tipo de licencia del software que incluyen o del soporte proporcionado por la propia distribución general Ubuntu. Las secciones son: *main*, *restricted*, *universe* y *multiverse* que se explican en el punto 6.

En el artículo se repasan los conceptos mas importantes acerca de la gestión de paquetes en Edubuntu en particular, y por extensión en Debian. Así mismo se estudia la herramienta básica para la gestión del software instalado en entorno gráfico y las diferentes opciones desde una terminal de texto.

2. ¿Qué es un paquete Debian?

Se puede definir un paquete como un conjunto de archivos y metadatos que describen dichos ar

Los metadatos[1] son la información que necesita el paquete así como la información del propio paquete.

Cualquier aplicación incluida en una distribución GNU/Linux debe estar en un paquete, bien junto con otras aplicaciones, o bien ella misma constituir un paquete propio. Por ejemplo, el navegador web Firefox tiene su propio paquete llamado **mozilla-firefox** . El metapaquete **ubuntu-mediapack**
(
deb http://www.xente.mundo-r.com/roax/archive feisty/
) contiene un conjunto de aplicaciones relativas a codecs, tipos de fuentes, plugin de Flash o plugins para Firefox, utilidades de compresión/descompresión de archivos
,...
que se han empaquetado conjuntamente bajo este nombre de metapaquete.

Existen 2 tipos de paquetes:

-

Binarios: contiene ejecutables, archivos de configuración, páginas man/info, información de copyright y documentación en general. Tienen la extensión .deb y se dese
mpaquetan con la
utilidad
dpkg
.

-

Fuentes: conjunto de archivos que definen el paquete. S e empaquetan /desempaquetan con **dpkg-source**
.

Los **paquetes binarios** .deb contienen los siguientes archivos:

- **debian-binary:** contiene laversión del archivo **.deb**
- **sección de control** del paquete (**control.tar.gz**) que contiene:

- archivo **control** (metadatos): dependencias del paquete, prioridad, mantenedor, arquitectura, conflictos, versión, md5sum,...
- scripts que se ejecutan antes o despues de instalar el paquete para, por ejemplo, detener los servicios de los paquetes que se actualizan hasta que su instalación o actualización se completa o ejecutan órdenes necesarias para empezar o reanudar un servicio una vez que el paquete ha sido instalado o actualizado, etc.

- **data.tar.gz :** contiene todos los archivos que se instalarán, con sus rutas de destino.

Los **paquetes fuente** contienen los siguientes archivos:

- **. dsc :** archivo de descripción del paquete. En general es información sobre el paquete.
- **. orig.tar.gz :** archivo fuente original.
- **.diff.gz :** archivo con los cambios de Debian sobre el código fuente original.

3. Dependencias

El funcionamiento de un paquete puede que dependa de otros. Es decir, su ejecución requiere la presencia en el sistema de otros paquetes.

Cada paquete tiene asociado un archivo **control** donde están documentadas las dependencias de dicho paquete.

Existen varios niveles de dependencia. Como ejemplo utilizamos los paquetes A y B y se pretende instalar el paquete A.

Puede ocurrir que:

- **A dependa de B**: es necesario instalar B.
- **A recomienda B**: no es necesario el paquete B pero es conveniente.
- **A sugiere B**: el paquete B contiene archivos que mejoran la funcionalidad de A.
- **A está en conflicto con B**: el paquete A no funciona si B está instalado.

- **A reemplaza B**: los archivos del paquete A sustituyen los de B.
- **A proporciona B**: los archivos del paqute B están incluidos en los de A.

Por ejemplo, si se quiere conocer las dependencias del paquete **Totem** (aplicación utilizada para la reproducción de audio y vídeo) ejecutar en una terminal:

#dpkg -s totem

Package: totem

Status: install ok installed

Priority: optional

Section: gnome

Installed-Size: 44

Maintainer: Ubuntu Desktop Team <ubuntu-desktop@lists.ubuntu.com>

Architecture: all

Version: 2.18.1-0ubuntu3

Depends : totem-gstreamer (>= 2.18.1-0ubuntu3) | totem-xine (>= 2.18.1-0ubuntu3)

Description: A simple media player for the Gnome desktop (dummy package)

La línea de dependencias se ha subrayado y se comprueba que la instalación de Totem requiere la instalación de los paquetes **totem-gstreamer** y **totem-xine**.

4. Nomenclatura de los paquetes .deb

Los nombres de los paquetes binarios Ubuntu siguen la siguiente convención:

<nombre>_<NúmeroDeVersión>-<NúmeroDeRevisiónUbuntu>.deb

donde:

<nombre> es el nombre del paquete

<NúmeroDeVersión> es el número de versión establecido por el desarrollador. No sigue un estandar.

<NúmeroDeRevisiónUbuntu> es el número de revisión que normalmente implica cambios en el archivo de control, los scripts de instalación o desinstalación, o en los archivos de configuración usados en el paquete.

.deb es la extensión utilizada para identificar los paquetes de Ubuntu.

Por ejemplo, la aplicación Totem se instala en el sistema mediante el paquete**:**

totem_2.18.1-0ubuntu3_all.deb

totem es el nombre del paquete

2.18.1 es el número de versión

0ubuntu3 es el número de revisión de Ubuntu

all indica que es válido para cualquier arquitectura

.deb la extensión de paquete Ubuntu

5. Secciones de paquetes en Edubuntu

Debian y Edubuntu difieren en la forma de agrupar los paquetes y nombrar las secciones o componentes. Así, desde Edubuntu las secciones disponibles son las siguientes:

1.
1. **Main**

La sección ***main*** contiene los paquetes que cumplen los requisitos de la licencia de Ubuntu y a

los que da soporte.

Esta sección incluye todos los paquetes necesarios para una distribución GNU/Linux de uso general. Además los paquetes de esta sección tienen ayuda técnica y actualizaciones de seguridad.

1.
1. **Restricted**

La sección ***restricted*** contiene paquetes soportados por Ubuntu por su importancia, pero que no están disponibles bajo ninguna licencia libre y no se pueden incluir en *main*
.

En esta sección se incluyen los paquetes relativos a los controladores de ciertas tarjetas gráficas, como nVIDIA, que son propieatrios. Existe ayuda pero es más limitada que en la sección *main*, ya que los desarrolladores no siempre tienen acceso al código fuente.

1.
1. **Universe**

La sección ***universe*** contiene gran cantidad de software que no tiene apoyo por parte de Ubuntu.

Esta es la sección que permite a los usuarios instalar cualquier programa en el sistema, y, tengan la licencia que tengan, se almacenan aparte de las secciones *main* y *restricted*.

1.
1. **Commercial**

Esta sección, como su nombre indica, contiene software comercial.

1.
1. **Multiverse**

La sección ***multiverse*** contiene paquetes sin soporte ya que no cumplen los requisitos del Software Libre.

6. Gestión de paquetes .deb: herramientas

Desde Edubuntu la gestión de paquetes se puede realizar de forma sencilla mediante la herramienta gráfica **Synaptic** que se explica en el punto 8.

Existen otro tipo de mecanismos que trabajan a mas bajo nivel y de los cuales sólo se explican **apt**
y
dpkg
. Su utilización es en modo terminal y, aunque el entorno es menos amigable, es conveniente conocer algunas de las opciones mas elementales ya que pueden ser de utilidad para el usuario en determinadas situaciones 'especiales'.

A continuación se explican las órdenes **dpkg** y **apt** con sus opciones mas habituales.

Orden dpkg

dpkg es una herramienta de bajo nivel que utiliza para instalar, eliminar y gestionar

Es decir, **dpkg** instala o elimina paquetes y maneja la base de datos (/**var**/**lib**/**dpkg**/) del sistema con la situación de paquetes indicando claramente los cambios.

Su ejecución requiere privilegios de administrador del sistema.

Consideraciones:

-

La instalación de un paquete **.deb** utilizando **dpkg** sólo se utiliza cuando se dispone del paquete en local, no a través de la red.

-

No resuelve las posibles dependencias que se puedan presentar en la instalación del paquete.

Las opciones mas importantes de **dpkg** son las siguientes:

Opción

Descripción

-i <paquete.deb>

instala un paquete .deb	3

-L <paquete>

lista los archivos de un paquete

-l

lista paquetes instalados

-l <paquete.deb>

información acerca de un paquete

-s <paquete>

información del paquete instalado (contenido archivo	control). Admite comodines.

-S <archivo>

buscar en qué paquete de los instalados está el archivo dado.

dpkg-reconfigure <paquete>

configura el paquete indicado

-r <paquete>

borra el paquete instalado

--purge <paquete>

borra el paquete y sus archivos de configuración.

-c <paquete.deb>

ver los archivos que contiene un paquete binario incluido el dueño, grupo y permisos.

Existen muchas mas opciones disponibles para dpkg. Remitimos al lector a la página de

manual de la orden **dpkg** (**$man dpkg**).

Ejemplos:

1. Instalar un paquete .deb: hay que recordar que se requiere, primero, descargar el paquete desde Internet.

#dpkg -i totem_2.18.1-0ubuntu3_all.deb

1. Mostrar los archivos que copia en el sistema la instalación de la aplicación Totem:

#dpkg -L totem

/.

/usr

/usr/share

/usr/share/doc

/usr/share/doc/totem

/usr/share/doc/totem/copyright

/usr/share/doc/totem/changelog.Debian.gz

2. Motrar los archivos del paquete binario de Totem con información complementaria:

#dpkg -c /var/cache/apt/archives/totem_2.18.1-0ubuntu3_all.deb

drwxr-xr-x root/root 0 2007-04-05 18:13 ./

drwxr-xr-x root/root 0 2007-04-05 18:13 ./usr/

drwxr-xr-x root/root 0 2007-04-05 18:13 ./usr/share/

drwxr-xr-x root/root 0 2007-04-05 18:13 ./usr/share/doc/

drwxr-xr-x root/root 0 2007-04-05 18:13 ./usr/share/doc/totem/

-rw-r--r-- root/root 440 2007-04-05 18:08 ./usr/share/doc/totem/copyright

-rw-r--r-- root/root 13872 2007-04-05 18:08 ./usr/share/doc/totem/changelog.Debian.gz

3. Conocer el nombre completo y versión de un paquete instalado en el sistema. Por ejemplo, para Totem ejecutar:

#dpkg -l | grep totem

ii libtotem-plparser1 2.18.1-0ubuntu3 Totem Playlist Parser library - runtime vers

ii totem 2.18.1-0ubuntu3 A simple media player for the Gnome desktop

ii totem-gstreamer 2.18.1-0ubuntu3 A simple media player for the Gnome desktop

ii totem-mozilla 2.18.1-0ubuntu3 Totem Mozilla plugin

La línea subrayada se corresponde con la aplicación Totem.

Orden apt

La orden apt(a dvanced p ackaging t ool, herra

Su ejecución requiere privilegios de administrador del sistema.

apt dispone de dos interfaces en línea de órdenes:

-

apt-get es la herramienta en línea de órdenes usada para el manejo de paquetes.

-

apt-cache realiza una serie de operaciones sobre la caché de paquetes de APT. Permite operaciones de búsqueda en la información de los paquetes.

Las opciones mas importantes de **apt-get** son las siguientes:

Opción	**Descripción**
update	Obtiene una nueva lista de paquetes actualizando las descripciones de los paquetes que hay en la base
upgrade	Actualiza el sistema con actualizaciones disponibles. No elimina paquetes previamente instalados, ni ins

dist-upgrade

actualiza todo entre ramas de desarrollo. Con -u vemos TODOS los paquetes a actualizar.

install

Instala un paquete y sus dependencias.

install --reinstall

Reinstala un paquete ya instalado.

install -f

Para solucionar problemas de dependencias de paquetes ya instalados o a medio instalar.

remove [--purge]

Elimina un paquete (borra archivos configuración).

source

Baja los paquetes fuente (los fuentes, los parches y los archivos de control de Debian) de un paquete d

clean

Los paquetes instalados se guardan en	**/var/cache/apt/archives/**

check

Comprobación de posibles problemas en el estado de los paquetes.

Ejemplos:

1. Actualizar el sistema (la lista de paquetes, no los propios paquetes):

#apt-get update

2. Instalar nuevas versiones de los paquetes ya instalados en el sistema:

#apt-get upgrade

3. Instalar un paquete desde la red resolviendo las dependencias que pudiera presentar:

#apt-get install totem

4. Tenemos un paquete instalado y queremos volver a instalarlo completamente:

#apt-get install --reinstall totem

5. El sistema presenta problemas de dependencias que han quedado pendientes de resolver y que impiden la instalación de nuevos paquetes:

#apt-get install -f

Las opciones mas importantes de **apt-cache** son las siguientes:

Opción	**Descripción**
show <paquete>	muestra la descripción del paquete dado

search <cadena>

busca la 'cadena' en la base de datos de paquetes

stats

muestra estadísticas sobre apt-cache

depends

muestra de qué paquetes depende otro paquete

showpkg

muestra inormación sobre el paquete

Ejemplos:

1. Queremos conocer información sobre un paquete que tenemos instalado:

#apt-cache show totem

2. Queremos conocer información sobre un paquete que no tenemos necesariamente instalado:

#apt-cache search dvdrip

3. Queremos conocer la relación de dependencias de un paquete dado:

#apt-cache depends dvdrip

7. El archivo sources.list

El archivo /**etc**/**apt**/**sources.list** indica a dónde se debe ir a buscar los paquetes que se instalan con la orden **apt-get install** , es decir, las fuentes o lugares que suministran toda las aplicaciones disponibles de la distribución.

Al instalar la variante Edubuntu Feisty Fawn de la distribución Ubuntu el sistema dispone del siguiente archivo /**etc**/**apt**/**sources.list** que reproducimos en parte:

 # more /etc/apt/sources.list

 # La sección main y restricted, fuentes y sus actualizaciones

deb http://es.archive.ubuntu.com/ubuntu/ feisty main restricted

deb-src http://es.archive.ubuntu.com/ubuntu/ feisty main restricted

deb http://es.archive.ubuntu.com/ubuntu/ feisty-updates main restricted

deb-src http://es.archive.ubuntu.com/ubuntu/ feisty-updates main restricted

 # Las secciones universe y multiverse y fuentes

deb http://es.archive.ubuntu.com/ubuntu/ feisty universe

deb-src http://es.archive.ubuntu.com/ubuntu/ feisty universe

deb http://es.archive.ubuntu.com/ubuntu/ feisty multiverse

deb-src http://es.archive.ubuntu.com/ubuntu/ feisty multiverse

 # Actualizaciones de seguridad y fuentes de todas las secciones

deb http://security.ubuntu.com/ubuntu feisty-security main restricted

deb-src http://security.ubuntu.com/ubuntu feisty-security main restricted

deb http://security.ubuntu.com/ubuntu feisty-security universe

deb-src http://security.ubuntu.com/ubuntu feisty-security universe

deb http://security.ubuntu.com/ubuntu feisty-security multiverse

deb-src http://security.ubuntu.com/ubuntu feisty-security multiverse

Conviene tener siempre el sistema actualizado con las últimas versiones de los paquetes instalados, siempre que correspondan a una versión estable de la distribución.

Tal y como se dijo en el punto 6 la orden **apt-get** permite, tanto tener actualizada la lista de paquetes disponibles en el sistema como su actualización a las últimas versiones.

Hay que tener en cuenta que hay que actualizar el sistema siempre que:

- Introducimos cambios en el archivo /**etc/apt/sources.list**
- Queremos actualizar[4] la lista de paquetes del sistema

Para ello hay que ejecutar la orden:

#apt-get update

Continua leyendo este artículo