



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Com crear els teus propis paquets Debian (6007 lectures)

Per **Toni Serna Rosselló**, [toniserna](http://informatica.escuelaedib.com) (<http://informatica.escuelaedib.com>)

Creado el 03/02/2009 12:31 modificado el 18/02/2009 13:24

El propòsit d'aquest tutorial és mostrar com distribuir els binaris de les teves aplicacions per a Linux amb els avantatges del sistema d'empaquetat Debian, i així poder instal·lar-les i desinstal·lar-les còmoda i fàcilment en un entorn més o menys privat. Dit d'una altra manera, aquest tutorial no cobreix com preparar paquets per a que formin part dels repositoris oficials, per això hauríeu d'estudiar la [Debian New Maintainers' Guide](#)⁽¹⁾ o visitar un excel·lent [tutorial](#)⁽²⁾ que trobareu a la secció "Enllaços" de Bulma.

Este artículo es una revisión y traducción del original en español que podéis consultar [aquí](#)⁽³⁾ si lo preferís.

Introducció

En un únic arxiu ".deb" podem empaquetar tots els arxius que formen el conjunt d'una aplicació o qualsevol altre col·lecció d'arxius que ens interressi distribuir. Dins un arxiu ".deb" també podem especificar les dependències obligatòries o recomanables que vinculen el nostre paquet amb altres, podem controlar on s'ha de copiar cada arxiu del nostre projecte i indicar quines operacions complementàries s'han de fer abans i després de la copia dels fitxers. A més els arxius distribuïts en forma de paquet Debian poden desinstal·lar-se amb una sola instrucció, sense deixar rastres i amb la possibilitat adicional d'especificar quines accions s'han d'efectuar abans i després del procés d'eliminació de fitxers.

Nocions generals abans de començar

En el directori `/var/cache/apt/archives` podrem trobar múltiples arxius ".deb" amb els que podrem fer qualche experiment.

Amb el comando

```
$ dpkg-deb -I nom_de_paquet.deb
```

obtindrem una descripció general del que ofereix un paquet en particular.

```
$ dpkg-deb -c nom_de_paquet.deb
```

ens mostrarà la llista de tots els arxius que s'instal·laran amb un paquet determinat.

Estructura d'un paquet ".deb"

Per a veure què conté un paquet podem executar:

```
$ ar tv nom_de_paquet.deb
```

Per exemple:



```
$ ar tv pitivi_0.11.1-0ubuntu1_all.deb
```

ens mostra:

```
rw-r--r-- 0/0 4 Jan 2 22:47 2009 debian-binary
rw-r--r-- 0/0 3086 Jan 2 22:47 2009 control.tar.gz
rw-r--r-- 0/0 467447 Jan 2 22:47 2009 data.tar.gz
```

El fitxer **debian-binary** conté simplement el número de versió del format del paquet. Per exemple "2.0"

El **data.tar.gz** és el que conté tots els fitxers que s'instal·laran i a quins directoris.

Mentres que el fitxer **control.tar.gz** és un arxiu comprimit que inclou diversos arxius amb informació complementària. A continuació descriurem breument els arxius més importants que conté aquest últim.

Fitxers de control (continguts dins "control.tar.gz")

control

Aquest és l'únic que és estrictament imprescindible. Es tracta d'un fitxer de text que amb uns "tags" i un valor associat a cada un d'ells es detallen diversos aspectes que és necessari documentar de cada paquet Debian. S'entendrà millor amb un exemple que podem copiar i modificar per adaptar-lo als nostres paquets:

```
Package: hola
Priority: optional
Section: devel
Installed-Size: 45
Maintainer: El meu nom i correu-e
Architecture: i386
Version: 1.0-12
Depends: libc6 (>= 2.1)
Description: Suposem que he fet el típic "Hello world". Es tracta d'un programa molt simple però
que ens serveix d'exemple imaginari per parlar de paquets Debian.
.
Suposam que el programa es va escriure en ANSI C i que ara estam distribuïnt l'executable en forma
de paquete ".deb".
```

- "Package" especifica el "nom" del paquet. Amb aquest nom i les comandes "dpkg" el podrem manipular un cop instal·lat.
- La prioritat ("Priority") indica com d'important és el paquet per al sistema, amb aquesta informació programes semi-intel·ligents de gestió de paquets poden classificar el nostre en una categoria com ara "Optional". Altes valors possibles són: Essential, Required, Important, Standard i Extra.
- La secció ("Section") permet determinar on s'emmagatzemarà el nostre paquet en cas de pujar-lo a un FTP de Debian.
- "Installed-size" serveix perquè els programes d'instal·lació de paquets puguin anticipar quin espai ocuparà el que estam distribuïnt un cop al sistema i així poder determinar si hi ha espai lliure a bastament.
- "Maintainer" s'empra per a indicar el nom i l'adreça de correu electrònic de la persona responsable del manteniment d'aquest paquet.
- Declararem també a "Architecture" per a què plataforma hem desenvolupat la nostra aplicació. A l'exemple és per a i386, però pot ser amd64, alpha ... o "all" si es tracta d'un conjunt de "shell scripts" independents de la plataforma.
- La versió ("Version") és la declaració que fa el desenvolupador de l'estadi de desenvolupament d'aquest paquet. El darrer número ("-12") és el número de revisió.
- El tag "Depends" indica els paquets (i opcionalment les seves versions mínimes) que han de ser obligatoriament instal·lats per garantir el funcionament del nostre paquet. A més del tag "Depends" pot haver-n'hi d'altres com ara: Recommends, Suggests, Conflicts, Replaces y Provides per poder indicar respectivament paquets recomanats, suggerits, que entren en conflicte, que són substituïts per el nostre o de funcionalitat completament inclosa dins el nostre.



- Finalment "*Description*" permet incorporar una breu descripció general del paquet i de les seves característiques. Els distints paràgrafs s'han de separar amb una línia que només inlogui un "."

Aquests són els "tags" més importants que s'han d'incorporar a cada fitxer "control", però no es tracta d'una llista exhaustiva. Per a més informació consulta el [Debian Policy Manual](#)⁽⁴⁾.

postinst, prerm, preinst y postrm

Aquests quatre arxius són scripts executables de "bash" que es llançaran automàticament abans o després de les instal·lacions i desinstal·lacions. Han d'incloure comandes de consola vàlides i com qualsevol altre script han de començar amb una línia "#!/bin/bash". Tots ells són opcionals.

- "**postinst**". Es fa servir per a les operacions que volem que s'executin després d'haver copiat els arxius del paquet. En aquest script, per exemple, podem assignar permisos a alguns dels arxius que acabem d'instal·lar o donar avisos especials a l'usuari o llançar assistents de configuració del paquet.
- "**prerm**". S'executa quan hem sol·licitat desinstal·lar el nostre paquet i just abans de que s'esborrin els fitxers que el formen. Típicament s'empra per aturar dimonis associats al nostre paquet.
- "**preinst**". Aquest script s'executa abans de que el paquet es descomprimeixi i sol emprar-se per aturar serveis que seran actualitzats per la instal·lació del nostre paquet.
- "**postrm**". Típicament elimina arxius creats durant l'execució de la nostra aplicació i que volem eliminar en desinstal·lar-la o altres operacions similars com eliminar grups d'usuaris creats per a la gestió de la nostra aplicació.

Anem per feina

Ara crec que ja podem abordar un cas pràctic. Suposem que tenim una aplicació formada per dos arxius: "SAM" (un shell-script que vull instal·lar a /usr/utilitats) i ".samrc" (que ha d'anar a /etc/sam).

Imaginem que estic al mateix directori on tinc els arxius "SAM" i ".samrc" (/home/toniserna/trash_project per fixar idees).

Aquí mateix creo un directori anomenat "debian" i per davall d'aquest reproduïxo la estructura dels directoris on he de copiar els arxius a la màquina receptora. Es tracta de fer com si aquest directori "debian" representés el directorio "/" (arrel) de les màquines on vull instal·lar el meu paquet. En el nostre cas ho faré amb una sola instrucció:

```
$ mkdir -p debian/usr/utilitats debian/etc/sam
```

Copio els 2 fitxers del meu projecte als directoris corresponents

```
$ cp SAM debian/usr/utilidades
```

```
$ cp .samrc debian/etc/sam
```

Cream el nostre arxiu "control"

Dins el directori "debian" que ja tenim, hem de crear un altre subdirector anomenat "DEBIAN" (ara tot en majúscules) i a dins, amb un editor de text crearem un arxiu anomenat "control".

```
$ mkdir debian/DEBIAN
```

```
$ vi debian/DEBIAN/control
```

Abans ja hem comentat què hem de posar als tags d'aquest arxiu. Per poder especificar correctament les dependències de la nostra aplicació i si volem filar prim, ens pot ser útil saber que escrivint "dpkg -S fitxer" el sistema em dirà quins paquets es necessari que estiguin instal·lats al sistema per a que "fitxer" funcioni. Per exemple:

```
$ dpkg -S /bin/cat
```



```
coreutils: /bin/cat
```

Així sabem que tot programa que faci servir "cat" requereix que el paquet "coreutils" estigui instal·lat. Llavors amb

```
$ apt-cache showpkg coreutils
```

sabrem la versió de "coreutils" que tenim instal·lada en el nostre sistema.

Creació dels scripts complementaris: postinst | preerm | preinst | postrm

Amb el nostre editor preferit (Vi en el meu cas) crearem d'aquests 4 scripts els que necessitem per dur a terme operacions complementàries a la nostra instal·lació. En aquest exemple suposarem que necessito només "postinst" i "preerm". Aquests arxius els he de posar allà mateix on hem posat el fitxer "control": debian/DEBIAN

```
$ vi debian/DEBIAN/postinst
```

```
$ vi debian/DEBIAN/preerm
```

En aquests scripts hi heu de programar el que trobeu oportú, els provau, i un cop depurats ja teniu tot el material a punt per compilar el vostre paquet ".deb"

Ajustar els permisos dels directoris

Aquesta maniobra no sempre és necessària, però com que no costa gaire la feim i lestos:

```
$ find ./debian -type d | xargs chmod 755
```

Compilació

Basta executar:

```
$ fakeroot dpkg-deb --build debian
```

```
dpkg-deb: building package "hola" in "debian.deb"
```

Això haurà generat un arxiu "debian.deb". Li canviem el nom per un de més proper als estàndards:

```
$ mv debian.deb hola_1.0-12_i386.deb
```

Instal·lar i desinstal·lar

Per a instal·lar (recorda que has de ser root):

```
$ sudo dpkg -i hola_1.0-12_i386.deb
```

Més tard quan volguem desinstal·lar-lo:

```
$ sudo dpkg -r hola
```

Notau: que no vos passi desapercebut el fet de què per a instal·lar hem d'emprar el nom de l'arxiu ".deb" i per a desinstal·lar ens hem de referir al nom del paquet (que vé determinat pel tag "Package" a l'arxiu "control").

Conclusions

Amb les passes que hemos seguit tindrem un paquet completament operatiu que ens resultarà molt útil per a fer instal·lacions i desinstal·lacions molt còmodament. Val a dir que queden diverses coses que podríem millorar per fer un paquet de més qualitat, però la idea d'aquest article era fer-ho el més fàcil possible i no allargar-me massa. Per saber més vos recomano dos articles excel·lents: el [Debian Binary Package Building HOWTO](#)⁽⁵⁾, que ha sigut una de les



meves fonts principals i també el que ja he esmentat a la introducció: [Creació de paquets Debian \(CATALÀ\)](#)⁽⁶⁾ o [Creación de paquetes Debian \(ESPANYOL\)](#)⁽⁷⁾

Lista de enlaces de este artículo:

1. <http://www.debian.org/doc/maint-guide/>
 2. <http://www.bulma.net/body.phtml?nIdNoticia=2395>
 3. http://www.informatica.escuelaedib.com/index.php?option=com_myblog&show=CA-mo-cr
 4. <http://www.debian.org/doc/debian-policy/ch-controlfields.html>
 5. <http://tldp.org/HOWTO/Debian-Binary-Package-Building-HOWTO/>
 6. http://acacha.dyndns.org/mediawiki/index.php/Creaci%C3%B3_de_paquets_Debian
 7. http://acacha.dyndns.org/mediawiki/index.php/Creaci%C3%B3n_de_paquetes_Debian
-

E-mail del autor: toniserna_ARROBA_ yahoo.es

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2483>