Costruire pacchetti Debian

Giovanni Mascellani

GULP - Gruppo Utenti Linux di Pisa

giovedì 10 aprile 2008

Di pacchetti ce n'è di tanti tipi...

- Pacchetti binari:
  - I classici file con estensione `.deb`;
  - Sono quelli usati direttamente dagli utenti (generalmente tramite APT);
  - "Binari" non vuol dire necessariamente che contengono eseguibili compilati o altri file binari.

Di pacchetti ce n'è di tanti tipi...

- Pacchetti binari:
  - I classici file con estensione `.deb`;
  - Sono quelli usati direttamente dagli utenti (generalmente tramite APT);
  - "Binari" non vuol dire necessariamente che contengono eseguibili compilati o altri file binari.
- Pacchetti sorgente
  - Sono quelli su cui gli sviluppatori (oggi noi!) mettono le mani;
  - Vengono compilati per generare i pacchetti binari;
  - Un pacchetto sorgente può generare anche più di un pacchetto binario.

Pacchetti binari

Il nome del file: `pacchetto_vers_arch.deb`.

- `pacchetto`: piuttosto facile, è semplicemente il nome del pacchetto!

Pacchetti binari

Il nome del file: `pacchetto_vers_arch.deb`.

- `pacchetto`: piuttosto facile, è semplicemente il nome del pacchetto!
- `vers`: versione del pacchetto, che deve seguire determinati schermi di cui parleremo più tardi.

Pacchetti binari

Il nome del file: `pacchetto_vers_arch.deb`.

- `pacchetto`: piuttosto facile, è semplicemente il nome del pacchetto!
- `vers`: versione del pacchetto, che deve seguire determinati schermi di cui parleremo più tardi.
- `arch`: l'architettura per cui è compilato questo pacchetto (`all` se è installabile su qualsiasi architettura).

Breve dissezione di un pacchetto binario

I pacchetti binari non sono altro che file `ar`!

Breve dissezione di un pacchetto binario

I pacchetti binari non sono altro che file `ar`!

- `debian-binary`: contiene semplicemente la versione del formato del pacchetto (attualmente 2.0);

Breve dissezione di un pacchetto binario

I pacchetti binari non sono altro che file `ar`!

- `debian-binary`: contiene semplicemente la versione del formato del pacchetto (attualmente 2.0);
- `data.tar.gz`: i file contenuti nel pacchetto. Durante l'installazione del pacchetto sono copiati al loro posto nel filesystem;

Breve dissezione di un pacchetto binario

I pacchetti binari non sono altro che file `ar`!

- `debian-binary`: contiene semplicemente la versione del formato del pacchetto (attualmente 2.0);
- `data.tar.gz`: i file contenuti nel pacchetto. Durante l'installazione del pacchetto sono copiati al loro posto nel filesystem;
- `control.tar.gz`: un po' di file di controllo, come descrizione del pacchetto, dipendenze, script di installazione e disinstallazione, hash MD5 dei file contenuti nel pacchetto. Impareremo il loro significato più tardi.

Pacchetti sorgente

Sono composti da ben tre file separati (a volte due, nel caso di pacchetti nativi per Debian, ma noi non ce ne preoccuperemo):

- `pacchetto_vers.orig.tar.gz`: il codice sorgente originario, generalmente la copia esatta del file distribuito dall'autore del software.

Pacchetti sorgente

Sono composti da ben tre file separati (a volte due, nel caso di pacchetti nativi per Debian, ma noi non ce ne preoccuperemo):

- `pacchetto_vers.orig.tar.gz`: il codice sorgente originario, generalmente la copia esatta del file distribuito dall'autore del software.
- `pacchetto_vers.diff.gz`: una patch che viene applicata al sorgente originario, e che lo modifica in modo da renderlo compilabile alla *Debian way*. Questo file non esiste nei pacchetti nativi per Debian.

Pacchetti sorgente

Sono composti da ben tre file separati (a volte due, nel caso di pacchetti nativi per Debian, ma noi non ce ne preoccuperemo):

- `pacchetto_vers.orig.tar.gz`: il codice sorgente originario, generalmente la copia esatta del file distribuito dall'autore del software.
- `pacchetto_vers.diff.gz`: una patch che viene applicata al sorgente originario, e che lo modifica in modo da renderlo compilabile alla *Debian way*. Questo file non esiste nei pacchetti nativi per Debian.
- `pacchetto_vers.dsc`: un file, firmato da chi ha generato il pacchetto sorgente, che contiene gli hash MD5 degli altri due file insieme ad altre informazioni generali sul pacchetto.

Compilazione di un pacchetto

- Possiamo procurarci un pacchetto sorgente con il comando:

  ```
  $ apt-get source pacchetto
  ```

  Il pacchetto verrà scaricato, decompresso e gli verrà applicata la patch.

Compilazione di un pacchetto

- Possiamo procurarci un pacchetto sorgente con il comando:

  ```
  $ apt-get source pacchetto
  ```

  Il pacchetto verrà scaricato, decompresso e gli verrà applicata la patch.
- Poi lo compiliamo con

  ```
  $ cd pacchetto-vers/
  $ dpkg-buildpackage -us -uc
  ```

File creati durante la compilazione

Dopo la compilazione saranno comparsi nuovi file:

- I pacchetti binari, ovviamente!

File creati durante la compilazione

Dopo la compilazione saranno comparsi nuovi file:

- I pacchetti binari, ovviamente!
- Un file `.changes`, che viene passato ad un programma (`dput` o `dupload`) che si preoccupa di caricare i file necessari sul server dell'archivio che li dovrà ospitare. A noi stasera non interessa.

Una piccola premessa: la versione

```
[epoch:]upstream_version[-debian_revision]
```

Una piccola premessa: la versione

`[epoch:]upstream_version[-debian_revision]`

- `upstream_version`: la parte principale, ossia la versione del software originale che si sta pacchettizzando.

Una piccola premessa: la versione

`[epoch:]upstream_version[-debian_revision]`

- `upstream_version`: la parte principale, ossia la versione del software originale che si sta pacchettizzando.
- `debian_revision` *(quasi opzionale)*: il numero di revisione Debian, che viene incrementato ogni volta che si produce una nuova versione del pacchetto e viene riportato a 1 quando si pacchettizza una nuova versione upstream. Questa parte non esiste solo per i pacchetti nativi per Debian.

Una piccola premessa: la versione

`[epoch:]upstream_version[-debian_revision]`

- `epoch` *(opzionale)*: un numero che serve per ovviare a problemi di upload con versioni sbagliate o di cambiamento di schema delle versioni. Se manca si considera pari a zero.
- `upstream_version`: la parte principale, ossia la versione del software originale che si sta pacchettizzando.
- `debian_revision` *(quasi opzionale)*: il numero di revisione Debian, che viene incrementato ogni volta che si produce una nuova versione del pacchetto e viene riportato a 1 quando si pacchettizza una nuova versione upstream. Questa parte non esiste solo per i pacchetti nativi per Debian.

Debianizzazione iniziale

- Decomprimere la tarball originale. Tutti i suoi file devono stare in una directory che si chiama `pacchetto-vers/`.

Debianizzazione iniziale

- Decomprimere la tarball originale. Tutti i suoi file devono stare in una directory che si chiama `pacchetto-vers/`.
- Entrare nella directory ed eseguire:

```
$ cd pacchetto-vers/
$ dh_make -f ../original.tar.gz
```

Debianizzazione iniziale

- Decomprimere la tarball originale. Tutti i suoi file devono stare in una directory che si chiama `pacchetto-vers/`.
- Entrare nella directory ed eseguire:

```
$ cd pacchetto-vers/
$ dh_make -f ../original.tar.gz
```

- `dh_make` chiederà che tipo di pacchetto si vuole costruire e provvederà a creare uno schema base seconda la scelta fatta.

Effetti della debianizzazione

In particolare, `dh_make` si preoccupa di:

- Creare una copia della tarball originale con il nome impostato opportunamente.

Effetti della debianizzazione

In particolare, `dh_make` si preoccupa di:

- Creare una copia della tarball originale con il nome impostato opportunamente.
- Creare una sottodirectory `debian/` con tanti file che descrivono il pacchetto. In generale molti servono solo per determinati tipi di pacchetti, alcuni invece valgono per tutti e devono essere opportunamente modificati.

Effetti della debianizzazione

In particolare, `dh_make` si preoccupa di:

- Creare una copia della tarball originale con il nome impostato opportunamente.
- Creare una sottodirectory `debian/` con tanti file che descrivono il pacchetto. In generale molti servono solo per determinati tipi di pacchetti, alcuni invece valgono per tutti e devono essere opportunamente modificati.
  Gli altri file del pacchetto sorgente verranno creati automaticamente alla prima compilazione.

Principali file del sorgente

`debian/control`

- Contiene i parametri principali del pacchetto sorgente e di tutti i pacchetti binari che questo genera.

Principali file del sorgente

`debian/control`

- Contiene i parametri principali del pacchetto sorgente e di tutti i pacchetti binari che questo genera.
- È strutturato in campi, divisi in vari paragrafi separati da linee vuote (o con soli spazi e tabulazioni). Ciascun campo ha la forma

  ```
  Chiave: valore
  ```

Principali file del sorgente

`debian/control`

- Contiene i parametri principali del pacchetto sorgente e di tutti i pacchetti binari che questo genera.
- È strutturato in campi, divisi in vari paragrafi separati da linee vuote (o con soli spazi e tabulazioni). Ciascun campo ha la forma

  ```
  Chiave: valore
  ```

- Alcuni campi possono estendersi su più linee, nel qual caso le linee dopo la prima devono iniziare con uno spazio o una tabulazione.

Principali file del sorgente

`debian/control` (campi principali)

I più importanti campi di `debian/control` contengono informazioni a proposito di:

Principali file del sorgente

`debian/control` (campi principali)

I più importanti campi di `debian/control` contengono informazioni a proposito di:

- Per il pacchetto sorgente:

Principali file del sorgente

`debian/control` (campi principali)

I più importanti campi di `debian/control` contengono informazioni a proposito di:

- Per il pacchetto sorgente:
  - Nome del mantenitore e dei suoi collaboratori;

Principali file del sorgente

`debian/control` (campi principali)

I più importanti campi di `debian/control` contengono informazioni a proposito di:

- Per il pacchetto sorgente:
  - Nome del mantenitore e dei suoi collaboratori;
  - Sezione (`main`, `contrib` o `non-free`; `admin`, `devel`, ...);

Principali file del sorgente

`debian/control` (campi principali)

I più importanti campi di `debian/control` contengono informazioni a proposito di:

- Per il pacchetto sorgente:
  - Nome del mantenitore e dei suoi collaboratori;
  - Sezione (`main`, `contrib` o `non-free`; `admin`, `devel`, ...);
  - Priorità (`required`, `important`, `standard`, `optional`, `extra`);

Principali file del sorgente

`debian/control` (campi principali)

I più importanti campi di `debian/control` contengono informazioni a proposito di:

- Per il pacchetto sorgente:
  - Nome del mantenitore e dei suoi collaboratori;
  - Sezione (`main`, `contrib` o `non-free`; `admin`, `devel`, ...);
  - Priorità (`required`, `important`, `standard`, `optional`, `extra`);
  - Dipendenze e conflitti di compilazione

Principali file del sorgente

`debian/control` (campi principali)

I più importanti campi di `debian/control` contengono informazioni a proposito di:

- Per il pacchetto sorgente:
  - Nome del mantenitore e dei suoi collaboratori;
  - Sezione (`main`, `contrib` o `non-free`; `admin`, `devel`, ...);
  - Priorità (`required`, `important`, `standard`, `optional`, `extra`);
  - Dipendenze e conflitti di compilazione
- Per i pacchetti binari:

Principali file del sorgente

`debian/control` (campi principali)

I più importanti campi di `debian/control` contengono informazioni a proposito di:

- Per il pacchetto sorgente:
  - Nome del mantenitore e dei suoi collaboratori;
  - Sezione (`main`, `contrib` o `non-free`; `admin`, `devel`, ...);
  - Priorità (`required`, `important`, `standard`, `optional`, `extra`);
  - Dipendenze e conflitti di compilazione
- Per i pacchetti binari:
  - Nome del pacchetto;

Principali file del sorgente

`debian/control` (campi principali)

I più importanti campi di `debian/control` contengono informazioni a proposito di:

- Per il pacchetto sorgente:
  - Nome del mantenitore e dei suoi collaboratori;
  - Sezione (`main`, `contrib` o `non-free`; `admin`, `devel`, ...);
  - Priorità (`required`, `important`, `standard`, `optional`, `extra`);
  - Dipendenze e conflitti di compilazione
- Per i pacchetti binari:
  - Nome del pacchetto;
  - Descrizioni: una breve (che stia in circa 60 caratteri) ed una corta (più righe);

Principali file del sorgente

`debian/control` (campi principali)

I più importanti campi di `debian/control` contengono informazioni a proposito di:

- Per il pacchetto sorgente:
  - Nome del mantenitore e dei suoi collaboratori;
  - Sezione (`main`, `contrib` o `non-free`; `admin`, `devel`, ...);
  - Priorità (`required`, `important`, `standard`, `optional`, `extra`);
  - Dipendenze e conflitti di compilazione
- Per i pacchetti binari:
  - Nome del pacchetto;
  - Descrizioni: una breve (che stia in circa 60 caratteri) ed una corta (più righe);
  - Relazioni con altri pacchetti (dipendenze, raccomandazioni, suggerimenti, conflitti, ...);

Principali file del sorgente

`debian/control` (campi principali)

I più importanti campi di `debian/control` contengono informazioni a proposito di:

- Per il pacchetto sorgente:
  - Nome del mantenitore e dei suoi collaboratori;
  - Sezione (`main`, `contrib` o `non-free`; `admin`, `devel`, ...);
  - Priorità (`required`, `important`, `standard`, `optional`, `extra`);
  - Dipendenze e conflitti di compilazione
- Per i pacchetti binari:
  - Nome del pacchetto;
  - Descrizioni: una breve (che stia in circa 60 caratteri) ed una corta (più righe);
  - Relazioni con altri pacchetti (dipendenze, raccomandazioni, suggerimenti, conflitti, ...);
  - Architetture su cui il pacchetto può essere compilato (`all` se lo stesso pacchetto va bene per tutte le architetture, `any` se può essere compilato su qualsiasi architettura).

Principali file del sorgente

`debian/rules`

- Si preoccupa della compilazione effettiva del pacchetto

Principali file del sorgente

`debian/rules`

- Si preoccupa della compilazione effettiva del pacchetto
- Si tratta di un Makefile sotto mentite spoglie.

Principali file del sorgente

`debian/rules`

- Si preoccupa della compilazione effettiva del pacchetto
- Si tratta di un Makefile sotto mentite spoglie.
- Deve essere eseguibile ed iniziare con la riga

  `#!/usr/bin/make -f`

  (che permette di eseguirlo direttamente senza chiamare `make`).

Principali file del sorgente

`debian/rules` (regole necessarie)

Oltre a quelle elencate qui, `debian/rules` può contenerne altre a discrezione del mantenitore. In ogni caso, però, quelle qui riportate devono funzionare senza necessità di interazione con l'utente.

Principali file del sorgente

`debian/rules` (regole necessarie)

Oltre a quelle elencate qui, `debian/rules` può contenerne altre a discrezione del mantenitore. In ogni caso, però, quelle qui riportate devono funzionare senza necessità di interazione con l'utente.

- `build`: si preoccupa di compilare effettivamente il software contenuto nel pacchetto. Non può richiedere i privilegi di root.

Principali file del sorgente

`debian/rules` (regole necessarie)

Oltre a quelle elencate qui, `debian/rules` può contenerne altre a discrezione del mantenitore. In ogni caso, però, quelle qui riportate devono funzionare senza necessità di interazione con l'utente.

- `build`: si preoccupa di compilare effettivamente il software contenuto nel pacchetto. Non può richiedere i privilegi di root.
- `binary`, `binary-arch` e `binary-indep`: si preoccupano di costruire i pacchetti binari associati a questo sorgente. In particolare, `binary-arch` deve costruire i pacchetti dipendenti dall'architettura e `binary-indep` quelli indipendenti. `binary` tipicamente dipende dagli altri due. Devono essere invocati come root (o tramite programmi come fakeroot).

Principali file del sorgente

`debian/rules` (regole necessarie)

Oltre a quelle elencate qui, `debian/rules` può contenerne altre a discrezione del mantenitore. In ogni caso, però, quelle qui riportate devono funzionare senza necessità di interazione con l'utente.

- `build`: si preoccupa di compilare effettivamente il software contenuto nel pacchetto. Non può richiedere i privilegi di root.
- `binary`, `binary-arch` e `binary-indep`: si preoccupano di costruire i pacchetti binari associati a questo sorgente. In particolare, `binary-arch` deve costruire i pacchetti dipendenti dall'architettura e `binary-indep` quelli indipendenti. `binary` tipicamente dipende dagli altri due. Devono essere invocati come root (o tramite programmi come fakeroot).
- `clean`: deve semplicemente riportare lo stato del pacchetto a com'era prima di ogni `build` o `binary`. Deve essere invocato come root.

Principali file del sorgente

`debian/changelog`

- Non si limita a contenere informazioni sulla storia del pacchetto, ma indica anche la versione, la distribuzione e l'urgenza del caricamento (che ha senso soltanto per i pacchetti che entreranno in una distribuzione complessa).

Principali file del sorgente

debian/changelog

- Non si limita a contenere informazioni sulla storia del pacchetto, ma indica anche la versione, la distribuzione e l'urgenza del caricamento (che ha senso soltanto per i pacchetti che entreranno in una distribuzione complessa).
- È formato da tante stanze, ciascuna delle quali ha la seguente forma:

```
pacchetto (versione) distribuzione; urgency=urgenza

  * Una modifica. (closes: #numero_bug)
  * Altra modifica.
    + Una sottomodifica.
    + Altra sottomodifica.

 -- Nome Cognome <email>  data
```

Principali file del sorgente

debian/changelog

- Non si limita a contenere informazioni sulla storia del pacchetto, ma indica anche la versione, la distribuzione e l'urgenza del caricamento (che ha senso soltanto per i pacchetti che entreranno in una distribuzione complessa).
- È formato da tante stanze, ciascuna delle quali ha la seguente forma:

```
pacchetto (versione) distribuzione; urgency=urgenza

  * Una modifica. (closes: #numero_bug)
  * Altra modifica.
    + Una sottomodifica.
    + Altra sottomodifica.

 -- Nome Cognome <email>  data
```

- La data è formattata come riportata da `date -R`: `Tue, 01 Apr 2008 12:45:50 +0200`.

Principali file del sorgente

`debian/copyright`

- Contiene informazione sullo stato legale del pacchetto: copyright e licenza.

Principali file del sorgente

`debian/copyright`

- Contiene informazione sullo stato legale del pacchetto: copyright e licenza.
- Se intendete mettere il vostro pacchetto su Debian, deve essere fatto molto accuratamente, possibilmente controllando tutti i file del pacchetto (ci si può aiutare con `licensecheck`).

Principali file del sorgente

`debian/copyright`

- Contiene informazione sullo stato legale del pacchetto: copyright e licenza.
- Se intendete mettere il vostro pacchetto su Debian, deve essere fatto molto accuratamente, possibilmente controllando tutti i file del pacchetto (ci si può aiutare con `licensecheck`).
- Altrmenti ve lo potete gestire come volete voi, anche lasciarlo come il modello di `dh_make` o bianco. La responsabilità in merito alla distribuzione del pacchetto è tutta vostra!

Principali file del sorgente

Il resto...

Il resto dei file contenuti in `debian/` non hanno direttamente a che fare con il pacchetto, ma sono usati dalla collezione di programmi `debhelper`.

Principali file del sorgente

Il resto...

Il resto dei file contenuti in `debian/` non hanno direttamente a che fare con il pacchetto, ma sono usati dalla collezione di programmi `debhelper`.
Vediamoli direttamente in azione!

Link utili

http://www.debian.org/doc/maint-guide/ La Debian New Maintainer's Guide, il punto di partenza per iniziare a costruire pacchetti Debian e collaborare con il progetto.

http://www.debian.org/doc/debian-policy/ La Debian Policy, ossia le regole che i pacchetti devono seguire per poter essere inclusi in Debian. Esistono anche altre policy più piccole che riguardano specifiche categorie di pacchetti (Java, Python, Perl, ...).

http://www.debian.org/doc/developers-reference/ La Developer's Reference, ossia il manuale dello sviluppatore Debian per tutto ciò che non riguarda strettamente la pacchettizzazione: struttura del progetto, votazioni, best practices, gestione dei bug, tool utili, ...).

http://www.debian.org/devel/ L'angolo degli sviluppatori di Debian: il punto di partenza per cercare documentazione relativa al collaborare con Debian.