Debian Binary Package Building HOWTO

Chr. Clemens Lee

clemens@kclee.de

2002-11-30, $Date: 2005/08/10 14:03:35 $

Diario delle Revisioni

Revisione 4.0 2005-08-09 Revisionato da: ccl
aggiornati indirizzi email e aggiunto un link alla traduzione Turca da Oguz Yarimtepe

Revisione 2.0 2003-12-13 Revisionato da: ccl
applicati i consigli di Frank Lichtenheld

Revisione 1.0 2003-11-08 Revisionato da: ccl
prima versione

Questo mini-HOWTO mostra come costruire un pacchetto Debian .deb minimale. Traduzione a cura di Hugh Hartmann (hhartmann@libero.it) e revisione a cura di Claudio Cattazzo.

Sommario

1. Introduzione

L'intento voluto è quello di installare il pacchetto appena creato solo sul proprio sistema e non quello di farlo entrare nella distribuzione ufficiale Debian. Per seguire il processo di creazione 'ufficiale', si è pregati di studiare la Debian New Maintainers' Guide (http://www.debian.org/doc/maint-guide/).

I comuni pacchetti Debian hanno un proprio pacchetto sorgente che include il file debian/rules che automatizza i passaggi necessari alla creazione di un pacchetto binario. Qui mostriamo solo come pacchettare un semplice script di shell od un eseguibile dentro un piccolo pacchetto binario Debian.

Tra l'altro, assumo che si conosca l'uso di 'tar', 'man', cos'è un un file '.tar.gz' e cos'è Debian (http://www.debian.org/) ;-), ma assumo che non si siano mai toccati programmi come 'ar' o 'dpkg'.

1.1. Risorse sul web

La Debian Reference (La guida Debian) (http://www.debian.org/doc/manuals/reference/reference.en.html) [NdT: in italiano (http://www.debian.org/doc/manuals/reference/reference.it.html)] fornisce un'eccellente panoramica così come informazioni dettagliate per ogni argomento specifico di Debian.

Il documento ufficiale per la creazione dei propri pacchetti Debian è la Debian New Maintainers' Guide (http://www.debian.org/doc/maint-guide/).

2. Iniziare

Dalla Debian Reference 2.2.2 2002-11-30: "Il formato interno di questo pacchetto binario Debian è descritto nella pagina di manuale deb(5). Dato che questo formato interno è soggetto a cambiamenti (tra release maggiori di Debian), si usi sempre dpkg-deb(8) per la manipolazione dei file .deb."

Dalla pagina di manuale di dpkg-deb: "dpkg-deb pacchetta, spacchetta e fornisce informazioni riguardo gli archivi Debian. I file .deb possono essere manipolati anche solo con `ar` e `tar` se necessario. Si usi dpkg per installare e rimuovere i pacchetti dal proprio sistema."

Diversi esempi di file .deb si possono trovare nella directory '`/var/cache/apt/archives/`'. Con '`dpkg-deb -I nomepacchetto.deb`' si può ottenere una panoramica generale di cosa offre questo pacchetto in particolare. '`dpkg-deb -c nomepacchetto.deb`' elenca tutti i file che verranno installati.

Si ottenga l'elenco del contenuto del file .deb con '`ar tv nomepacchetto.deb`'. Si usi l'opzione 'x' per estrarre i file.

3. Struttura dei pacchetti

Esaminiamo un esempio di pacchetto un po' più da vicino. Per esempio il file '`parted_1.4.24-4_i386.deb`' contiene questi tre file:

```
$ ar tv parted_1.4.24-4_i386.deb
rw-r--r-- 0/0      4 Mar 28 13:46 2002 debian-binary
rw-r--r-- 0/0   1386 Mar 28 13:46 2002 control.tar.gz
rw-r--r-- 0/0  39772 Mar 28 13:46 2002 data.tar.gz
```

Ora possiamo iniziare ad estrarre tutti i file incluso il contenuto dei file tar.

3.1. debian-binary

Il contenuto di questo file è "2.0\n". Questo stabilisce la versione del formato del file .deb. Per il formato 2.0 tutte le altre righe vengono ignorate.

3.2. data.tar.gz

Il file 'data.tar.gz' contiene tutti i file che saranno installati con i loro percorsi di destinazione:

```
drwxr-xr-x root/root         0 2002-03-28 13:44:57 ./
drwxr-xr-x root/root         0 2002-03-28 13:44:49 ./sbin/
-rwxr-xr-x root/root     31656 2002-03-28 13:44:49 ./sbin/parted
drwxr-xr-x root/root         0 2002-03-28 13:44:38 ./usr/
drwxr-xr-x root/root         0 2002-03-28 13:44:41 ./usr/share/
drwxr-xr-x root/root         0 2002-03-28 13:44:38 ./usr/share/man/
drwxr-xr-x root/root         0 2002-03-28 13:44:52 ./usr/share/man/man8/
-rw-r--r-- root/root      1608 2002-03-28 13:44:37 ./usr/share/man/man8/parted.8.gz
drwxr-xr-x root/root         0 2002-03-28 13:44:41 ./usr/share/doc/
drwxr-xr-x root/root         0 2002-03-28 13:44:52 ./usr/share/doc/parted/
-rw-r--r-- root/root      1880 2002-03-07 14:20:08 ./usr/share/doc/parted/README.Debian
-rw-r--r-- root/root      1347 2002-02-27 01:40:50 ./usr/share/doc/parted/copyright
-rw-r--r-- root/root      6444 2002-03-28 13:37:33 ./usr/share/doc/parted/changelog.Debian.gz
-rw-r--r-- root/root     15523 2002-03-28 02:36:43 ./usr/share/doc/parted/changelog.gz
```

Deve essere l'ultimo file nell'archivio deb.

3.3. control.tar.gz

Nel nostro esempio questo file ha il seguente contenuto:

```
-rw-r--r--    1 root     root         1336 Mar 28  2002 control
-rw-r--r--    1 root     root          388 Mar 28  2002 md5sums
-rwxr-xr-x    1 root     root          253 Mar 28  2002 postinst
-rwxr-xr-x    1 root     root          194 Mar 28  2002 prerm
```

'md5sums' contiene, per ogni file in data.tar.gz, l'md5sum. Nel nostro esempio il contenuto appare come questo:

```
1d15dcfb6bb23751f76a2b7b844d3c57  sbin/parted
4eb9cc2e192f1b997cf13ff0b921af74  usr/share/man/man8/parted.8.gz
2f356768104a09092e26a6abb012c95e  usr/share/doc/parted/README.Debian
a6259bd193f8f150c171c88df2158e3e  usr/share/doc/parted/copyright
7f8078127a689d647586420184fc3953  usr/share/doc/parted/changelog.Debian.gz
98f217a3bf8a7407d66fd6ac8c5589b7  usr/share/doc/parted/changelog.gz
```

Non bisogna preoccuparsi, il file 'md5sum' così come i file 'postinst' e 'prerm' non sono necessari per il proprio primo pacchetto. Ma è meglio ricordarsi della loro esistenza, ogni pacchetto che appartiene alla Debian ufficiale ha questi file per delle buone ragioni.

I file 'prerm' e 'postinst' sembra che si prendano cura della rimozione dei file della vecchia documentazione e aggiungano un link da doc a share/doc.

```
$ cat postinst
#!/bin/sh
set -e
# Aggiunto automaticamente da dh_installdocs
if [ "$1" = "configure" ]; then
  if [ -d /usr/doc -a ! -e /usr/doc/parted -a -d /usr/share/doc/parted ]; then
    ln -sf ../share/doc/parted /usr/doc/parted
  fi
fi
# Fine della sezione aggiunta automaticamente

$ cat prerm
#!/bin/sh
set -e
# Aggiunto automaticamente da dh_installdocs
if [ \( "$1" = "upgrade" -o "$1" = "remove" \) -a -L /usr/doc/parted ]; then
  rm -f /usr/doc/parted
fi
# Fine della sezione aggiunta automaticamente
```

Ed infine il file più interessante:

```
$ cat control
Package: parted
Version: 1.4.24-4
Section: admin
Priority: optional
Architecture: i386
Depends: e2fsprogs (>= 1.27-2), libc6 (>= 2.2.4-4), libncurses5 (>= \
5.2.20020112a-1), libparted1.4 (>= 1.4.13+14pre1), libreadline4 (>= \
4.2a-4), libuuid1
Suggests: parted-doc
Conflicts: fsresize
Replaces: fsresize
Installed-Size: 76
Maintainer: Timshel Knoll <timshel@debian.org>
Description: Il programma per il ridimensionamento delle partizioni del disco GNU Parted
 Parted è un programma che permette di creare, eliminare, ridimensionare,
 muovere e copiare partizioni del disco rigido. È utile per creare dello
 spazio per nuovi sistemi operativi, riorganizzare l'uso del disco, e per
 copiare i dati in nuovi dischi rigidi.
 .
 Questo pacchetto contiene il binario di Parted e la pagina di manuale.
 .
 Parted attualmente supporta etichette/tavole delle partizioni di DOS, Mac,
 Sun, BSD, GPT e PC98, così come un tipo di 'loop' (raw disk) che ne
 permette l'utilizzo su RAID/LVM. I filesystem supportati sono: ext2, ext3,
 FAT (FAT16 e FAT32) e linux-swap. Parted può anche identificare filesystem
 HFS (Mac OS), JFS, NTFS, ReiserFS, UFS e XFS ma non può ancora
 creare/rimuovere/ridimensionare/verificare questi filesystem.
```

```
 .
 La natura di questo software è tale che ogni baco potrebbe causare una
 massiccia perdita di dati. Nonostante non ci siano bachi conosciuti al
 momento, potrebbero esisterne, quindi si è pregati di fare un back up di
 tutti i file importanti prima di eseguire questo programma, sapendo di
 agire a proprio rischio.
```

Ulteriori informazioni riguardo il file control possono essere ottenute con '`man 5 deb-control`'.

4. Fare pratica

Ora è il momento di fare pratica da soli. Ho un semplice script di shell chiamato 'linuxstatus' che desidero installare come '/usr/bin/linuxstatus'. Quindi si crei in primo luogo una directory chiamata 'debian' e si copi, all'interno di questa, il file 'linuxstatus'.

```
$ mkdir -p ./debian/usr/bin
$ cp linuxstatus ./debian/usr/bin
```

4.1. control

Iniziamo con il file control. Il numero di versione deve avere un trattino seguito da un numero che descrive la versione del pacchetto Debian, per esempio '1.1-1'. Se il proprio programma consiste solo di uno script shell portabile, si usi 'all' come sua 'Architecture'.

Per il campo 'Depends' è necessario scoprire a quale pacchetto appartiene un determinato file o programma dal quale il proprio nuovo pacchetto dipende. Per scoprire questo si può usare 'dpkg -S <file>', p.e..:

```
$ dkpg -S /bin/cat
coreutils: /bin/cat
```

Quindi per scoprire di più circa il pacchetto 'coreutils' si può usare il comando 'apt-cache showpkg coreutils' che dirà, tra le altre cose, il numero di versione attualmente installato sul sistema.

Come nota a margine, ci sono altri due maniere per trovare le stesse informazioni. C'è una pagina web dove si possono cercare file Debian: http://www.debian.org/distrib/packages. Si vada in fondo alla pagina e si riempia il form.

Ultima ma non per importanza, c'è una graziosa applicazione GUI chiamata 'kpackage', che fornisce comode opzioni di navigazione tra i pacchetti e permette anche di ricercare pacchetti, dati i nomi dei singoli file.

I campi 'Suggests', 'Conflicts', 'Replaces', ecc. possono essere omessi se non necessari.

Ecco quindi il risultato del nostro primo file 'control':

```
Package: linuxstatus
Version: 1.1-1
Section: base
Priority: optional
Architecture: all
Depends: bash (>= 2.05a-11), textutils (>= 2.0-12), awk, procps (>= \
1:2.0.7-8), sed (>= 3.02-8), grep (>= 2.4.2-3), coreutils (>= 5.0-5)
```

```
Maintainer: Chr. Clemens Lee <clemens@kclee.de>
Description: Informazioni sul sistema Linux
 Questo script fornisce una vasta descrizione
 dei differenti aspetti del sistema.
```

Il file ’control’ viene copiato in una directory chiamata ’DEBIAN’ all’interno dell’altra directory ’debian’.

```
$ mkdir -p debian/DEBIAN
$ find ./debian -type d | xargs chmod 755   # questo è necessario su Debian Woody, non chiedetemi pe
$ cp control debian/DEBIAN
```

Se ci si aspetta che il proprio pacchetto abbia in futuro un’utenza maggiore, può essere d’aiuto la lettura di questo articolo: Writing Debian package descriptions (http://people.debian.org/~walters/descriptions.html).

4.2. dpkg-deb

Ora è quasi pronto. Si dia:

```
$ dpkg-deb --build debian
dpkg-deb: building package ‘linuxstatus’ in ‘debian.deb’.
$ mv debian.deb linuxstatus_1.1-1_all.deb
```

È stato più facile di quanto previsto. Ora è sufficiente installare questo pacchetto sul proprio sistema ed è tutto fatto:

```
root# dpkg -i ./linuxstatus_1.1-1_all.deb
```

Si scriva ’`linuxstatus`’ o ’`ls -l /usr/bin/linuxstatus`’ per vedere se ha funzionato. Se non si gradisce più il proprio pacchetto, si scriva solo ’`dpkg -r linuxstatus`’ e si controlli nuovamente se il pacchetto è stato disinstallato. Comunque, se si installa una nuova versione del pacchetto non si deve rimuovere prima il pacchetto precedente.

Se si è curiosi riguardo allo schema del numero della versione e le convenzioni usate per nominare un pacchetto Debian, si legga this section (http://www.debian.org/doc/manuals/reference/ch-system.en.html#s-pkgname) in The Debian Reference (http://www.debian.org/doc/manuals/reference/reference.en.html) [NdT: in italiano (http://www.debian.org/doc/manuals/reference/ch-system.it.html#s-pkgname)].

5. Doppia Verifica

Ora che si è ottenuta una prima impressione e si è costruito il proprio pacchetto binario, è tempo di diventare un pò più "seri" e dare uno sguardo alla qualità del pacchetto che abbiamo prodotto.

5.1. lintian

Fortunatamente per noi il progetto Debian fornisce uno strumento per verificare i pacchetti Debian. Questo strumento si chiama ’lintian’. Se non è ancora stato installato sul proprio sistema questo è il momento buono per farlo (`apt-get install lintian`).

Ora useremo questo piccolo prezioso strumento sul nostro nuovo pacchetto:

```
$ lintian linuxstatus_1.1-1_all.deb
E: linuxstatus: binary-without-manpage linuxstatus
E: linuxstatus: no-copyright-file
W: linuxstatus: prerm-does-not-remove-usr-doc-link
W: linuxstatus: postinst-does-not-set-usr-doc-link
```

Il pacchetto non sembra così perfetto. Mancano la pagina di manuale, il file di copyright e anche gli script 'prerm' e 'postinst'.

5.2. Documentazione Minima

Questo non è il luogo adatto per dire molto riguardo la scrittura e la creazione delle pagine di manuale, ci sono molti libri che hanno uno o più capitoli relativi a questo argomento e c'è anche online The Linux MAN-PAGE-HOWTO (http://tldp.org/HOWTO/Man-Page/index.html) [NdT: in italiano (http://it.tldp.org/HOWTO/Man-Page.html)]. Quindi ora si assume che ci sia una pagina di manuale perfetta per il proprio script alla locazione `./man/man1/linuxstatus.1`.

Lo stesso per il file 'copyright'. Si possono trovare diversi esempi nella directory `/usr/share/doc` con questo comando: `find /usr/share/doc -name "copyright"`.

Ecco quindi il nostro esempio di un file 'copyright':

```
linuxstatus

Copyright: Chr. Clemens Lee <clemens@kclee.de>

2002-12-07

La pagina principale di linuxstatus è su:
http://www.kclee.de/clemens/unix/index.html#linuxstatus

L'intero codice base può essere distribuito sotto i termini della GNU
General Public License (GPL), che appare immediatamente sotto. In
alternativa, tutto il codice sorgente come ogni codice derivato da questo
codice può essere distribuito invece sotto la GNU Lesser General Public
License (LGPL), a scelta del distributore. Il testo completo della LGPL
appare di seguito in questo file.

Si veda /usr/share/common-licenses/(GPL|LGPL)
```

Per gli script 'prerm' e 'postinst' si copino uno ad uno gli esempi (#postint) dal pacchetto 'parted' precedente al posto dei file aventi lo stesso nome nella propria directory del progetto. Questi file dovrebbero funzionare bene.

Si crei ora il pacchetto debian nuovamente. Nel file 'control' si aumenti il numero della versione da 1.1-1 a 1.2-1 (quando si scriverà una nuova pagina di manuale si aumenterà il proprio numero di release interno). È anche necessario copiare i nuovi file nelle loro destinazioni appropriate:

```
$ mkdir -p ./debian/usr/share/man/man1
$ mkdir -p ./debian/usr/share/doc/linuxstatus
$ find ./debian -type d | xargs chmod 755
$ cp ./man/man1/linuxstatus.1 ./debian/usr/share/man/man1
$ cp ./copyright ./debian/usr/share/doc/linuxstatus
```

```
$ cp ./prerm ./postinst ./debian/DEBIAN
$ gzip --best ./debian/usr/share/man/man1/linuxstatus.1
$
$ dpkg-deb --build debian
dpkg-deb: building package ‘linuxstatus’ in ‘debian.deb’.
$ mv debian.deb linuxstatus_1.2-1_all.deb
```

Gzip è necessario perché lintian si aspetta che i file delle pagine di manuale siano compressi il più possibile.

5.3. fakeroot

Ora vediamo se il pacchetto è diventato un buon "cittadino" di Debian:

```
$ lintian linuxstatus_1.2-1_all.deb
E: linuxstatus: control-file-has-bad-owner prerm clemens/clemens != root/root
E: linuxstatus: control-file-has-bad-owner postinst clemens/clemens != root/root
E: linuxstatus: bad-owner-for-doc-file usr/share/doc/linuxstatus/ clemens/clemens != root/root
E: linuxstatus: bad-owner-for-doc-file usr/share/doc/linuxstatus/copyright clemens/clemens != root/r
E: linuxstatus: debian-changelog-file-missing
```

Uff, ancora reclami. Ok, non rinunceremo. Attualmente la maggior parte degli errori sembra in realtà essere lo stesso problema. I nostri file sono tutti pacchettati per l’utente e il gruppo ’clemens’, mentre si assume che la maggior parte della gente preferisca averli installati come ’root/root’. Ma questo problema è facilmente superato usando lo strumento ’**fakeroot**’. Quindi correggiamo e verifichiamolo velocemente (mentre si ignora il risultato del file changelog):

```
$ fakeroot dpkg-deb --build debian
dpkg-deb: building package ‘linuxstatus’ in ‘debian.deb’.
$ mv debian.deb linuxstatus_1.2-1_all.deb
$ lintian linuxstatus_1.2-1_all.deb
E: linuxstatus: debian-changelog-file-missing
```

Bene, ma si deve ancora aggiungere un altro file al pacchetto.

5.4. Ulteriore Documentazione

Lasciatemi già dire che accanto al file ’changelog’ nella directory ’doc/linuxstatus’ è necessario un file ’changelog.Debian’. Entrambi dovrebbero essere compressi con gzip.

Di seguito ci sono due file di esempio, ’changelog’:

```
linuxstatus (1.2-1)

  * Made Debian package lintian clean.

 -- Chr. Clemens Lee <clemens@kclee.de>  2002-12-13
```

e ’changelog.Debian’:

```
Il manutentore di linuxstatus di Debian e
l’autore sono identici. Si veda quindi anche un file normale di changelog
per i cambiamenti in Debian.
```

La Debian Policy fornisce più dettagli riguardo il formato del file changelog (http://www.debian.org/doc/debian-policy/ch-miscellaneous.html#s-dpkgchangelog).

Ora si spera che l'ultimo passo sia:

```
$ cp ./changelog ./changelog.Debian ./debian/usr/share/doc/linuxstatus
$ gzip --best ./debian/usr/share/doc/linuxstatus/changelog
$ gzip --best ./debian/usr/share/doc/linuxstatus/changelog.Debian
$ fakeroot dpkg-deb --build ./debian
dpkg-deb: building package 'linuxstatus' in 'debian.deb'.
$ mv debian.deb linuxstatus_1.2-1_all.deb
$ lintian linuxstatus_1.2-1_all.deb
```

Ah, non si ottengono più reclami :-). Ora, come root si può installare questo pacchetto sopra a quello precedente, ancora con il solito comando '`dpkg -i`'.

```
root# dpkg -i ./linuxstatus_1.2-1_all.deb
(Reading database ... 97124 files and directories currently installed.)
Preparing to replace linuxstatus 1.1-1 (using linuxstatus_1.2-1_all.deb) ...
Unpacking replacement linuxstatus ...
Setting up linuxstatus (1.2-1) ...
```

6. Sommario

Per non creare confusione vengono riassunti tutti i passaggi che sono stati fatti per costruire il proprio pacchetto binario Debian.

File indispensabili:

1. uno o più file binari eseguibili o script di shell
2. una pagina di manuale per ogni file eseguibile
3. un file 'control'
4. un file 'copyright'
5. un file 'changelog' ed uno 'changelog.Debian'

Si impostino le sottodirectory temporanee di 'debian':

1. si crei la directory '`debian/usr/bin`' (o ovunque si pensi di mettere i propri file eseguibili)
2. si crei '`debian/usr/share/man/man1`' (o dentro qualsiasi sezione a cui la propria man page appartenga)
3. si crei la directory '`debian/DEBIAN`'
4. si crei '`debian/usr/share/doc/<nome_pacchetto>`'
5. ci si assicuri che tutte le sottodirectory di '`debian`' abbiano i permessi dei file a 0755

Si copino i file dentro l'albero delle directory temporaneo di 'debian':

1. si copi il file eseguibile nella directory '`debian/usr/bin`' (o ovunque si pensi di mettere i propri file eseguibili)

2. si copi il file della pagina di manuale dentro la directory '`debian/usr/share/man/man1`'
3. si copi il file 'control' dentro la directory '`debian/DEBIAN`'
4. si copino i file 'copyright', 'changelog' e 'changelog.Debian' dentro la directory '`debian/usr/share/doc/<nome_pacchetto>`'
5. si comprimano con gzip la pagina di manuale, i file 'copyright', 'changelog' e 'changelog.Debian' con l'opzione '--best' all'interno dell'albero delle directory temporaneo '`debian`'

Si costruisca e si verifichi il pacchetto binario Debian:

1. si invochi '`dpkg-deb --build`' usando '`fakeroot`' sulla directory '`debian`'
2. si rinomini il file risultante '`debian.deb`' con il suo nome finale includendo le informazioni della versione e dell'architettura
3. si verifichi il pacchetto .deb risultante in conformità con la Debian policy usando '`lintian`'

7. Quello che rimane

Ci sono molti dettagli che non sono stati affrontati in questo testo, ad esempio come distribuire demoni di Unix, file di configurazione e molto altro.

Ma la cosa più importante che voglio ancora mettere in evidenza è che per i manutentori Debian i pacchetti sono pacchetti sorgente, non pacchetti binari. Non interagiscono mai direttamente con l'interno dei pacchetti binari. Infatti, solo gli sviluppatori di '`dpkg-deb`' e '`dpkg`' hanno la necessità di sapere cosa sono. Infatti non è consigliato fare così.

Se uno sviluppatore dovesse spiegare a qualcuno come costruire un pacchetto Debian, spiegherebbe certamente come realizzare un pacchetto sorgente e come compilarlo.

D'altra parte, non tutti gli sviluppatori vogliono inviare il proprio software a Debian (per il momento), ma vogliono comunque approfittare dei vantaggi che un sistema di gestione dei pacchetti come 'dpkg' offre senza rilasciare il pacchetto sorgente. Personalmente rilascerò i miei progetti ancora come file tar.gz con il codice sorgente per tutti i generi di piattaforma, mentre progetto di offrire molti pacchetti '.deb' per la convenienza degli utenti di Debian che vogliono installare ed usare il mio software.

Se qualcuno vuole fare il passo successivo per sottoporre un pacchetto software a Debian deve mettersi prima a studiare la "Debian New Maintainers' Guide" così come il Debian Policy Manual (http://www.debian.org/doc/debian-policy/). Nella propria impresa di creare un pacchetto sorgente Debian, bisognerebbe dare uno sguardo anche alla mailing list debian-mentors (http://lists.debian.org/debian-mentors/) per vedere sviluppatori Debian principianti ed esperti interagire tra loro e affrontare problemi simili che si potrebbero incontrare.

8. Ringraziamenti

Si ringraziano

- Colin Watson per aver contribuito ad una frase nell'abstract ed aver fornito consigli riguardanti la struttura, gli obiettivi ed il titolo di questo HowTo,

- Bill Allombert per aver contribuito ad una frase nella sezione 'Quello che rimane' ed aver fornito consigli che rafforzano quello che Colin aveva già detto,
- Santiago Vila per la puntualizzazione che 'md' è un alias locale e che il mondo (Unix) si è standardizzato su 'mkdir',
- Tabatha Marshall dal progetto TLDP (http://www.tldp.org/) per il supporto generale,
- Joey Hess, Carlo Perassi e Joe Riel per correzioni minori,
- Claudio Cattazzo per la conversione del mio documento xml privato in un formato standard docbook assieme a vari minori miglioramenti e correzioni [2003-10-04].
- Frank Lichtenheld per aver trovato errori introdotti quando le pagine HTML vengono generate e per avere suggerito di usare 'dpkg -S <file>' quando si cerca un pacchetto contenente un certo file [2003-12-13].
- Hugh Hartmann per aver tradotto questo HOWTO in italiano (http://it.tldp.org/HOWTO/Debian-Binary-Package-Building-HOWTO/index.html) e Claudio Cattazzo per aver revisionato la traduzione [2003-12-13].
- Oguz Yarimtepe per la traduzione questo HOWTO in Turco (http://www.belgeler.org/howto/dpkg-deb-nasil.html) [2005-08-09].

9. Link

Debian New Maintainers' Guide:	http://www.debian.org/doc/maint-guide/
Debian:	http://www.debian.org/
[NdT: in italiano:	http://it.tldp.org/HOWTO/Debian-Binary-Package-Building-HOWTO/index.html]
The Debian Reference:	http://www.debian.org/doc/manuals/reference/reference.en.html
[NdT: in italiano:	http://www.debian.org/doc/manuals/reference/reference.it.html]
Pacchetti Debian:	http://www.debian.org/distrib/packages
Writing Debian package descriptions:	http://people.debian.org/~walters/descriptions.html
The Linux MAN-PAGE-HOWTO:	http://www.tldp.org/HOWTO/mini/Man-Page.html
[NdT: in italiano:	http://it.tldp.org/HOWTO/Man-Page/index.html]
Debian Policy Manual:	http://www.debian.org/doc/debian-policy/
Debian Mentors:	http://lists.debian.org/debian-mentors/
The Linux Documentation Project:	http://www.tldp.org/
Plug: la mia piccola pagina su Unix:	http://www.kclee.de/clemens/unix/