

Il pacchetto perfetto (Aggiornato)

Per creare il pacchetto perfetto,, dobbiamo seguire alcune regole. Alcune scritte, altre no. Di seguito, ne vedremo alcune, nella speranza che le seguitate, per creare i vostri pacchetti.

- [Nome del pacchetto](#)
- [Cartelle di installazione](#)
- [Il file slack-desc](#)
- [Documenti](#)
- [Altro](#)
- [Aggiunte opzionali](#)

Nome del pacchetto

Potrete domandarvi come mai il nome del pacchetto è così importante. Bene, dice molto riguardo al pacchetto in questione. Prima di tutto, dal nome del file, possiamo vedere la piattaforma per la quale è stato compilato, il nome e la versione del programma contenuto, la versione della "build", e chi l'ha compilata. Considerateli un po' come campi di una tabella. Abbiamo la parte con il nome, quella con la versione, quella con l'architettura, e quella con l'identificativo della "build"

apache-1.2.1-i386-2jim.tgz

Nell'esempio qui sopra, potete vedere prima di tutto che si tratta di Apache (campo nome), e più precisamente della versione 1.2.1 (campo versione). Possiamo inoltre vedere che è stato compilato e ottimizzato per la piattaforma intel 386 (campo architettura). I valori accettati per questo campo sono:

i386

i486

i586

i686

noarch (significa che non c'è una specifica piattaforma)

Poiché Slackware non supporta al momento altre piattaforme (quali potrebbero essere le sparc), vi basteranno queste 5 possibilità. E' bene evitare di usare il carattere "-" nel campo della versione, in quanto questo rende difficoltoso l'aggiornamento da parte del gestore di pacchetti, nel momento in cui si trovasse a cercare di determinare il numero della versione del pacchetto, confondendone il contenuto con gli altri campi del nome del pacchetto.

Il campo successivo, è quello della "build". Di solito sta a significare quante volte il pacchetto è stato compilato. Questo aiuta il gestore di pacchetti di Slackware a distinguere pacchetti diversi che però contengono lo stesso programma alla stessa versione. Nel caso del nostro esempio, si tratta della seconda "build". Questo potrebbe indicare un bugfix, o qualche cambiamento da quando la -1 è stata rilasciata, e questo pacchetto è quindi un aggiornamento della -1.

Viene inoltre aggiunta, di solito, alla fine del numero di versione della "build", una serie di tre lettere, che l'autore utilizza per distinguere i suoi pacchetti. Nel caso del nostro esempio, si tratta di "jim", che sta per Jim Simmons. Questo campo è importante per alcuni gestori di pacchetti, in quanto aiuta a distinguere le versioni ufficiali dei pacchetti, da quelle non ufficiali.

Cartelle di installazione

In questa breve guida, non copriremo tutte le cartelle di installazione possibili. Potete ad ogni modo trovarne una spiegazione più approfondita nell'area degli howto. Detto semplicemente, dovreste creare, all'interno del pacchetto, almeno quanto segue:

`./` (senza la quale il pacchetto non si rimuover correttamente)

`/install`

`/install/slack-desc` `/install/slack-required` (questo è opzionale, lo vedremo più tardi)

`/usr/doc/package-version/`

Ovviamente avrete anche le cartelle nelle quali il pacchetto si andrà a installare. Nella cartella `doc`, potrete mettere una copia della licenza, e tutta quella buona documentazione, come i file `README` ecc ecc.

Segue un elenco delle principali cartelle, correlato di una breve descrizione del loro utilizzo:

`/etc`

Questa parte contiene di solito i files di configurazione. Il comando di configurazione `--sysconfdir=/etc` viene utilizzato per assicurare che questi files finiscano proprio nella cartella in questione. Se lanciate `./configure --prefix=/usr` e il pacchetto ha dei files di configurazione, questi andranno generalmente a finire in `/usr/etc` il che non è esattamente di nostro gradimento. Può invece capitare, a volte, che vogliate mettere i files di configurazione in una sottocartella di `/etc`, come `/etc/mioprogramma`. Ad esempio, questo accade per i pacchetti di `gtk` e `gnome`.

`/usr`

Questo è il cuore vero e proprio del sistema, e dove la maggior parte degli eseguibili dovrebbe essere installata. In molti casi, usando `prefix=/usr` otterrete direttamente l'effetto desiderato.

`/var`

La parte `/var` è destinata ai files, i log, i pids, a tutto ciò che deve essere scritto dal programma. Il prefisso di configurazione per questa parte, è `./configure --localstatedir=NOME-CARTELLA`.

Vi starete ora domandando il perchè di tutta questa attenzione a dove il pacchetto e i suoi files di supporto verranno installati. Le ragioni sono molte, ma la principale è quella della standardizzazione. Un altro buon esempio è quello di avere la possibilità di far girare la gran parte del sistema in sola lettura. Se si seguono gli standard, questo può essere fatto molto facilmente. Basterà infatti montare alcune parti in `"rw"`, e il cuore del sistema in `"ro"`. Vogliamo davvero fornire un pacchetto standard ai nostri utenti, e alcune volte questo necessita qualche piccolo ritocco agli standard di `"make install"`. Ancora una volta, passando i corretti argomenti a `./configure` e usando qualcosa come `checkinstall`, avremo come risultato, nel 99% dei casi, un buon pacchetto.

Il file "slack-desc"

Il file `slack-desc` è molto importante in quanto aiuta a tenere aggiornato il database con buone descrizioni riguardo i pacchetti installati. Aiuta inoltre alcuni tool ad identificare correttamente i pacchetti. Per un approfondimento riguardo a questo file, leggete [The Slack-desc file](#). Abbiamo inoltre un [tool automatizzato](#) che vi aiuterà a creare un perfetto file `"slack-desc"`.

Documenti

Praticamente qualsiasi tipo di licenza prevede che una copia di se stessa vada allegata al programma distribuito. Questo è il principale motivo per cui abbiamo bisogno della cartella `"/usr/doc/package"`. Dopo tutto si tratta di software libero nella maggior parte dei casi, e ci sono quindi ben poche regole da seguire. Questa cartella dovrebbe inoltre contenere alcune informazioni sull'utilizzo del programma, se queste sono allegate ai sorgenti. E' anche un buon posto per semplici file di configurazione, che il programma dovrebbe utilizzare. La documentazione può anche essere in `/usr/share/doc/packages`, visto che questo non è altro che un link simbolico a `/usr/doc`. Alcuni programmi installano la documentazione lì, e questo è accettabile.

Altro

Ci sono alcuni accorgimenti che vi aiuteranno a creare il pacchetto perfetto. Prima di tutto, non ottimizzate la vostra compilazione per piattaforme i686. Vogliamo che i nostri pacchetti possano essere utilizzati da tutti, e non tutti hanno dei i686. Per il momento, la piattaforma di default per Slackware, è la i486. Per di più, il solo fatto che state compilando su un i686 non rende i vostri pacchetti ottimizzati per i686. Dovete specificarlo manualmente a gcc, per ottimizzare la compilazione. Molte persone fanno confusione su questo argomento. Un'ultima cosa, riguardo all'ottimizzazione, è da dire. Nella maggior parte dei casi, non ci saranno differenze visibili tra un i486 e un i686, anzi, ci sono migliaia di prove in rete che mostrano come a volte questa ottimizzazione potrebbe persino rallentare l'applicazione.

Ora, per creare il pacchetto perfetto, seguite i suggerimenti fin'ora suggerite. Poi riservate qualche minuto supplementare, per fare quanto segue:

comprimete con *gzip* le pagine di man

comprimete con *gzip* le pagine di informazione

eliminate tutti i symbols da binari e librerie con *strip*

controllate permessi e proprietà dei files che intendete includere nel pacchetto

Aggiunte Opzionali

Ci sono due principali tool per l'aggiornamento, in rete, ed entrambi presentano un supporto delle dipendenze. Swaret lo implementa con un controller interno che usa un file centrale sul sito web di swaret stesso. Per questo, Swaret non necessita alcuna modifica al pacchetto per supportarlo. Ad ogni modo, se volete che il vostro pacchetto si possa usare con ogni tool, dovrete apportare una lieve modifica al vostro prodotto. Un primo modo per segnalare le dipendenze è quello di elencare le dipendenze del pacchetto nello `slack-desc` file e sul sito web quando inviate il pacchetto. Si può altrimenti aggiungere al pacchetto un file chiamato `"slack-required"`, che dovrà essere inserito nella cartella `"/install"`. Avrete realmente bisogno di farlo solo se il programma che state pacchettizzando necessita di librerie o applicazioni che non sono installate nella Slackware standard, o che sono di una versione successiva rispetto a quella installata.

La struttura del file `"slack-required"` prevede una sola dipendenza per linea, e riportata secondo il seguente formato:

nome_applicazione

oppure

[nome_applicazione] [condizione] [versione]

dove il campo [condizione] potrà essere

=, =, =<, <, o >

<= e =< dovrebbero funzionare correttamente, in caso di errori di trascrizione, ad esempio

man >= 1.5l-i386-1

Notate che anche nel controllo abbiamo incluso l'architettura e il numero della "build", in modo da aiutare il controllo della versione.

Voi che siete i creatori del pacchetto dovrete sapere meglio di chiunque altro che cosa il pacchetto richiede per funzionare. E ricordate che dovrete compilare il pacchetto su un'installazione pulita di Slackware, e non certo sulla -current, o usando dropline gnome, altrimenti il vostro pacchetto non funzionerà sicuramente su un'altra macchina, a meno che voi non siate certi di ciò che il pacchetto richiede.

Abbiamo deciso di aggiungere questa parte dopo aver ricevuto numerose emails riguardanti gli script di compilazione.

Poichè al momento non esiste una posizione standard in cui inserire gli script di compilazione, abbiamo chiesto alla comunità di inserire questi script nella cartella "/usr/src/slackbuilds". Se avete uno script di compilazione che usate per creare il pacchetto, includetelo nel file che distribuirete, posizionandolo in "/usr/src/slackbuilds/nome_pacchetto"

Poichè si tratta generalmente di piccoli script, non appesantiranno eccessivamente il pacchetto, bensì contribuiranno a rendere più completo il pacchetto creato.

Questo è tutto riguardo la creazione del pacchetto perfetto. Scrivetemi due righe se ritenete che ci sia qualcosa da aggiungere.

Questo testo è frutto della traduzione di Giacomo Rizzo. Il testo originale è reperibile all'indirizzo <http://www.linuxpackages.net/howto.php?page=perfect-packageckage>. Per qualsiasi errore di traduzione, o modifica da apportare al testo tradotto, potete scrivere a [alt\[AT\]free-os.it](mailto:alt[AT]free-os.it)