

「Debian パッケージの作り方 (4) – ライブラリ編 –」

大浦 真 (ohura@debian.org)

1 はじめに

テーマ ライブラリの Debian パッケージ作成を実演してみる。普通のパッケージよりも少し複雑。

素材 libhello (“Hello, library world.” と表示する関数 hello を提供するライブラリ)

2 基本的な手順

1. 必要なパッケージのインストール。
 - **build-essential**
 - そのソフトウェアのビルドに必要なパッケージ。
 - **dh-make**、**debhelper**、**devscripts**、**fakeroot**、**lintian/linda**
2. パッケージを作りたいソフトウェアのアーカイブを用意し展開。
3. `dh_make` でソースツリーの雛型を作成。
4. ひとまず `debuild` でパッケージを作ってみる。
5. うまくいかない場合は、ログや `debian/rules` ファイルを確認。
6. うまくできた場合は、`lintian` の出力や `debian/` 以下のファイルを確認し、さらに修正。

3 ライブラリパッケージの作り方

- ライブラリパッケージは、共有ライブラリ本体が含まれるパッケージ (**libhello0**) と開発用のファイルが含まれるパッケージ (**libhello-dev**) に分かれている。
- ライブラリの場合も、`dh_make` で雛型を作ることができるがそのままではパッケージができない。
- `debian/control` の修正
 - `dh_make` の指示通り、`‘libhelloBROKEN’` となっている部分を修正。この場合は、`‘libhello0’` にする。
- `debian/rules` の修正
 - `‘$(MAKE) distclean’` の行を `‘[! -f Makefile] || $(MAKE) distclean’` に変更。
 - `dh_install` に `‘-sourcedir=debian/tmp’` というオプションを付ける必要がある*¹。
 - `dh_makeshlibs` を有効にする。-V オプションを付ける。有効にすると、適切な `postinst/postrm` スクリプトが用意される。(install 時と remove 時に `ldconfig` が実行される。)
- その他
 - `libhello1.dirs` と `libhello1.install` を `libhello0` に名前を変える。

*¹ かつては `dh_movefiles` というものがあったが、現在は使われていない。

- libhello-dev.install から **pkg-config** 関係を削除。
- 不要な debian/*.ex を削除。

4 共有ライブラリの名前

- 共有ライブラリは、三つの名前を持つ。

soname (/usr/lib/libhello.so.0): 実行ファイルが参照する名前。ライブラリのインターフェイスが変更されると末尾の数字が変わる。Debian ではこの数字をパッケージ名に付加する。

real name (/usr/lib/libhello.so.0.0.0): ライブラリの実体。soname はこのファイルへのシンボリックリンクになっている。

linker name (/usr/lib/libhello.so): ライブラリを利用するプログラムをコンパイルするときに参照する名前。

5 参考文献

- <http://www.linux.or.jp/JF/JFdocs/Program-Library-HOWTO/> 「Program Library HOWTO」(JF)
- <http://www.netfort.gr.jp/~dancer/column/libpkg-guide/libpkg-guide.html> 「Debian Library Packaging guide」

6 サンプル

libhello.c

```
#include <stdio.h>

void hello(void) {
    printf("Hello, library world.\n");
}
```

libhello.h

```
void hello(void);
```

demo_use.c

```
#include "libhello.h"

int main(void) {
    hello();
    return 0;
}
```