



RPM パッケージ作成ハンズオン 入門編

—第2回 Vine Linux ユーザの集い—

山本 宗宏 <munepi@vinelinux.org>

2010年5月29日@SRA OSS トレーニングルーム

produced by Project Vine and exciting guys

概 要

RPM パッケージになっていないサンプルソフトウェアを通して、RPM パッケージの作成を体験していただくセミナーです。入門編では、RPM パッケージ作成に関わるパッケージのビルドやテスト、vbuilder を使った src.rpm のビルドテストなどの一連の作業を学べます。

入門編の流れ

入門編の進行は、スライドにそって解説しながら1つ1つ作業をしていただきました。以下のスライドのpdf も合わせてご参考下さい。

<http://trac.vinelinux.org/repos/people/munepi/documents/10/vlm2ndb.pdf>

1. 準備
2. サンプルソフトウェアのコンパイル
3. 雛形 spec ファイルの編集
4. パッケージのビルド
5. パッケージのインストール、アンインストール
6. パッケージのビルドテスト

1 準備

手元に Vine Linux 5.1 がインストールされた計算機を用意して下さい。このハンズオンでは、Vine Linux 5.1 i386 を前提として解説しています。もし Vine Linux 5.1 ppc, x86_64 を用意している方は、適時読み替えてコマンドを実行して下さい。

1.1 ~/.rpmmacros の編集

RPM パッケージを作成するための最低限の設定をしましょう。適当なテキストエディタを用いて、`~/.rpmmacros` を開いて下さい。

```

1 %_topdir ${HOME}/rpm
2 # %_sysconfdir /etc
3
4 # %packager Your Name <your mail address>
5 # %packager      Munehiro Yamamoto <munepi@vinelinux.org>
6 %packager      munepi
7
8 # gpg signing
9 # %_signature gpg
10 # %_gpg_name Your Name <your mail address>
11
12 # more useful macros
13 %_sourcedir      %{_topdir}/SOURCES/%{name}
14 # %_builddir     /dev/shm/%(whoami)/BUILD
15 # %_buildrootdir /dev/shm/%(whoami)/BUILDROOT

```

- `%packager` : あなたの ID (半角英数字の文字列) を入力して下さい。あなたのハンドルネームやニックネームで構いません。特に、そのような ID をお持ちでない方や分からない方は、上記の例を参考に、名前とメールアドレスを入力して下さい。
- 今回は作成していただいた RPM パッケージに GPG 署名をしません。
- `%_sourcedir %{_topdir}/SOURCES/%{name}` : パッケージごとにソースやパッチなどの格納ディレクトリを設けます。パッケージごとにディレクトリを分けておくと、複数のパッケージをメンテナンスするときに、それぞれのファイルたちが混ざりません。Vine Linux 5 のデフォルトでは、`%_sourcedir %{_topdir}/SOURCES` となっています。
- `%_builddir /dev/shm/%(whoami)/BUILD`, `%_buildrootdir /dev/shm/%(whoami)/BUILDROOT` : それぞれ `rpmbuild` コマンドからビルドするときの作業ディレクトリ, `rpmbuild` コマンドからビルド済みのソフトウェアを作業フォルダにインストールするディレクトリです。4GB 以上のメモリがある場合には、`/dev/shm` 以下に作業ディレクトリを設けてもよいでしょう。

1.2 sudo の設定

`sudo` は一般ユーザに特定のコマンドを root 権限で実行させるためのコマンドです。su コマンドから root になってもよいのですが、今回は積極的に `sudo` を使います。

あなたはおそらく手元の計算機の管理者です。sudo をインストールします。

```
1 $ su -c "apt-get install sudo"
```

あなたのアカウントを `sudoers` に追加しましょう。

```
1 $ su -c /usr/sbin/visudo
```

vi の操作に不馴れな方は、環境変数 `EDITOR` に `gedit` などの使いなれたテキストエディタを代入して、以下を実行して下さい。

```
1 $ EDITOR=gedit su -c /usr/sbin/visudo
```

以下の例を参考にしながら、“## Allow root to run any commands anywhere” と書かれた行の下に、自分のアカウントを追加して保存します。

```
1 ## Allow root to run any commands anywhere
2 root    ALL=(ALL)    ALL
3 munepi  ALL=(ALL)    ALL
```

2 パッケージングしてみよう

2.1 サンプルを動かす

今回の RPM パッケージハンズオンのために、echowlvl というサンプルソフトウェアを用意しました。このサンプルソフトウェアを RPM パッケージにしてみましょう。

2.1.1 サンプルをダウンロード

Fx for Vine Linux や Chromium などの Web ブラウザで以下の URL に一旦アクセスしてから、echowlvl-0.1.tar.gz をダウンロードして下さい。

<http://trac.vinelinux.org/repos/people/munepi/documents/10/echowlvl-0.1.tar.gz>

2.1.2 サンプルをコンパイル

ソフトウェアをビルドするための最小限の開発環境を整えます。

```
1 $ sudo apt-get install build-essential
```

先ほどダウンロードしたサンプル echowlvl-0.1.tar.gz を展開して、“いつもの呪文”（./configure && make && make install のこと）を唱えてみましょう。

```
1 $ tar zxvf echowlvl-0.1.tar.gz
2 $ cd echowlvl-0.1
3 $ ./configure
4 $ make
5 $ ./echowlvl
```

コンパイルしてできた実行ファイル echowlvl を実行すると、どうなったでしょうか？ みなさんで一斉に叫んでみましょう！

2.2 spec ファイルを書く

echowlvl が “いつもの呪文” を唱えてビルドできたので、RPM パッケージを作成するためのソースコードともいえるべき spec ファイルを書きましょう。

2.2.1 雛形 spec ファイルのダウンロード

echowlvl の RPM パッケージ作成にすぐさまとりかかれるように、雛形を用意しました。Web ブラウザで以下の URL に一旦アクセスしてから、echowlvl-v1.spec をダウンロードして下さい。

<http://trac.vinelinux.org/repos/people/munepi/documents/10/echowlvl-v1.spec>

ダウンロードした `echowlvl-vl.spec` を `~/rpm/SPECS/` へ移動します。

```
1 $ mv echowlvl-vl.spec ~/rpm/SPECS/
```

2.2.2 spec ファイルの構成

spec ファイルは主に 4 つの部分から構成されています。

1. パッケージ情報
2. スクリプト
3. ファイルリスト
4. 更新履歴

各部を順番に見ながら埋めていきましょう。

■ **パッケージ情報** パッケージの基本情報はすべてここに凝縮されています。特にパッケージ依存関係は、`rpm` や `apt` に正しい情報を伝えるために重要になります。

```
1 ## -*- coding: utf-8-unix -*-
2 ## This is a template spec file for vlm2nd.
3 Name:          NAME
4 Version:       VERSION
5 Release:       1%{?_dist_release}
6 Summary:
7 Summary(ja):
8 Group:
9 License:
10 URL:           http://trac.vinelinux.org/wiki/OfflineMeeting/20100529
11 Source0:       http://trac.vinelinux.org/repos/people/munepi/documents
                  /10/{name}-{version}.tar.gz
12
13 BuildRoot:     %{_tmppath}/{name}-{version}-{release}-root
14
15 Distribution:  YOUR LINUX
16 Vendor:        YOUR PROJECT
17 Packager:      YOUR NAME (or YOUR ID)
18
19 %description
20
21 %description -l ja
```

- **Name** : パッケージ名です。 `{name}` というマクロで参照できます。今回は、`echowlvl` として下さい。
- **Version** : ソフトウェアのバージョンです。 `{version}` というマクロで参照できます。今回は、`0.1` として下さい。
- **Release** : リリース番号と Vine Linux のバージョンを組み合わせで書きます。リリース番号は、同じバージョンで何回目のリリースになるかを意味します。 `{?_dist_release}` は、Vine Linux のバージョンに置き換えられます。 `1%{?_dist_release}` は `1v15` となります。 `{release}` というマクロで参照できます。

- `Summary, Summary(ja)` : パッケージの概要です。パッケージの概要をそれぞれ英語と日本語で記入して下さい。
- `Group` : ソフトウェアの種類です。今回は `Applications/Other` にします。
- `License` : ソフトウェアのライセンスです。今回は `GPLv3+` にします。
- `%description, %description -l ja` : `Summary` よりも詳しいパッケージの解説を書きます。

■ **スクリプト** ソフトウェアのソースを展開し、必要であればパッチを当て、コンパイル、インストールの一連の作業をすべてここに記述します。

```

1 %prep
2 %setup -q
3
4 %build
5 %configure
6 %__make %{?_smp_mflags}
7
8 %install
9 %__rm -rf $RPM_BUILD_ROOT
10 %__make install DESTDIR=$RPM_BUILD_ROOT
11
12 %clean
13 %__rm -rf $RPM_BUILD_ROOT

```

“いつもの呪文”をざっくり spec ファイルに記述すると、このようになります。echowlvl の rpm 化はこれで十分なので、今回はこのままにしておきます。

■ **ファイルリスト** 実際にインストールされるファイルリストを書きます。

```

1 %files
2 %defattr(-,root,root,-)

```

さて、何を書けばよいのでしょうか？ 一旦、保留にしておきます。

■ **更新履歴** 更新履歴は、パッケージのメンテナだけでなく、パッケージに興味のある人が見るかもしれません。他の人が見てよいように、できるだけ分かりやすく記述するように心掛けましょう。

あなたの名前とメールアドレスを埋めて下さい。

```

1 %changelog
2 * Sat May 29 2010 YOUR NAME <YOUR E-MAIL ADDRESS> 0.1-1
3 - first release
4
5 ## end of file

```

2.3 パッケージのビルド

さきほどダウンロードした `echowlvl-0.1.tar.gz` を `~/rpm/SOURCES/echowlvl` に移します。

```

1 $ mkdir ~/rpm/SOURCES/echowlvl

```

```
2 $ mv echowlvl-0.1.tar.gz ~/rpm/SOURCES/echowlvl/
```

とりあえず、`rpmbuild -ba` でパッケージをビルドしてみます。

```
1 $ rpmbuild -ba ~/rpm/SPECS/echowlvl-v1.spec
```

そういえば、`%files` のファイルリストを埋めていませんでした。早速、`/usr/bin/echowlvl` をファイルリストに追加しましょう。

```
1 %files
2 %defattr(-,root,root,-)
3 %doc AUTHORS COPYING ChangeLog INSTALL NEWS README
4 %{_bindir}/echowlvl
```

`%doc` に入れたドキュメントファイルは、`/usr/share/doc/{name}-{version}` に格納されます。

再び、`rpmbuild -ba` で `echowlvl` パッケージをビルドして下さい。今度は `echowlvl` パッケージができあがっていませんか？

2.4 パッケージのインストール

`echowlvl` パッケージができたら、インストールをします。`echowlvl` パッケージが期待した通りにインストールできて、動作することを確認します。

```
1 $ sudo apt-get install ~/rpm/RPMS/i386/echowlvl-0.1-1v15.i386.rpm
```

`echowlvl` コマンドを実行して動作確認をしましょう。先ほどを同じ挙動をしていますでしょうか？

2.5 パッケージのアンインストール

今度は、`echowlvl` パッケージをアンインストールしてみます。

```
1 $ sudo apt-get remove echowlvl
```

3 さらにパッケージングしてみよう

3.1 サンプルが更新されたようです！

Web ブラウザで以下の URL に一旦アクセスしてから、`echowlvl-0.2.tar.gz` をダウンロードして下さい。

<http://trac.vinelinux.org/repos/people/munepi/documents/10/echowlvl-0.2.tar.gz>

新しくなった `echowlvl` をコンパイルして、動作することを確認して下さい。

`echowlvl` の新しいバージョンに合わせて `spec` ファイルを編集して下さい。`rpmbuild -ba` でパッケージのビルドを試みて下さい。新しいバージョンに対応した `echowlvl` パッケージはできましたでしょうか？ `echowlvl` パッケージができた場合は、パッケージのインストール、テスト、アンインストールもお忘れなく！

3.2 ビルドに必要なライブラリを追加

もし echowlvl のパッケージが作成できなかった方は、エラーメッセージを見ながら、原因を追究してみましょう。

今回は、curses.h というヘッダファイルがコンパイルに必要なだったようです。/usr/include/curses.h は、ncurses-devel パッケージに格納されています。

apt-get を使って ncurses-devel パッケージをインストールします。

```
1 $ sudo apt-get install ncurses-devel
```

再び、rpmbuild -ba でパッケージのビルドを試みて下さい。今度は echowlvl パッケージはできましたでしょうか？ echowlvl パッケージができた場合は、パッケージのインストール、テスト、アンインストールもお忘れなく！

4 ビルドテストしましょう

パッケージができれば、パッケージのビルドテストをしましょう。いつでもどんな環境でもパッケージングできた方が嬉しいですね。「あの人の環境ではビルドできるけど、私の環境ではビルドできないよ?」、「私の環境でもビルドできたけど、この機能が使えないよ?」といったことが起こると不都合ですよ。

パッケージのビルドテストには、vbuilder を使います。vbuilder は、(chroot という) 子環境に Vine Linux の最小環境から構築し、パッケージの依存関係などのチェックをしてからパッケージのビルドするツールです。

vbootstrap, vbuilder に関する詳細は、以下の URL に記載されています。

<http://trac.vinelinux.org/wiki/VineBootstrap>

4.1 vbuilder

vbuilder は vbootstrap パッケージに格納されています。

```
1 $ sudo apt-get install vbootstrap
```

4.1.1 vbuilder の基本操作

基本的な流れは以下の通りです。

```
1 $ sudo vbuilder clean      (過去に構築した子環境を削除)
2 $ sudo vbuilder build     (子環境に最小環境を構築)
3 $ sudo vbuilder build-rpm hoge-1.0.src.rpm
```

vbuilder にこれらのアクションを一度に渡すこともできて、以下のようにすれば、順次 clean → build → build-rpm のように実行します。

```
1 $ sudo vbuilder clean build build-rpm hoge-1.0.src.rpm
```

この場合、build を省略できます。

```
1 $ sudo vbuilder clean build-rpm hoge-1.0.src.rpm
```

なお、`$ sudo vbuilder --version 5.1 clean build` を実行すると、約 600 MB 程度の容量を消費します。vbuilder を実行する際には、ハードディスクの容量にご注意下さい。

4.2 src.rpm を vbuilder に投げてみる

echowlvl-0.2-1vl5.i386.rpm といっしょに、もう 1 つ echowlvl-0.2-1vl5.src.rpm という rpm ができています。src.rpm は、ソフトウェアのソースアーカイブやパッチファイルなど、spec ファイルをまとめたソースパッケージです。echowlvl-0.2-1vl5.src.rpm から echowlvl-0.2-1vl5.i386.rpm をビルドできます。

vbuilder に echowlvl-0.2-1vl5.src.rpm を投げてみましょう。一度 vbuilder を実行すると、一連の作業が終わるまでしばらくかかります。果たして vbuilder は echowlvl-0.2-1vl5.i386.rpm を返してくれるのでしょうか？

```
1 $ sudo vbuilder --version 5.1 clean \
2   build-rpm ~/rpm/SRPMS/echowlvl-0.2-1vl5.src.rpm
```

4.3 依存関係を解決する

vbuilder に echowlvl-0.2-1vl5.src.rpm を投げたら、“curses.h がありません”と言われてしまいました。echowlvl-0.2-1vl5.src.rpm はソースパッケージとして不完全のようです。

/usr/include/curses.h は ncurses-devel パッケージに格納されています。ncurses-devel パッケージを事前にインストールしていると、echowlvl-0.2-1vl5.src.rpm がビルドできそうです。早速、試してみましょう。

```
1 $ sudo vbuilder --version 5.1 clean \
2   install-rpm ncurses-devel \
3   build-rpm ~/rpm/SRPMS/echowlvl-0.2-1vl5.src.rpm
```

echowlvl-vl.spec を修正しましょう。Release はリリース番号を 2 にして下さい。BuildRequires: ncurses-devel を追加します。%changelog に BuildRequires: ncurses-devel を追加したことを記します。

```
1 Release:          2%{?_dist_release}
2 ...
3 Source0:          http://trac.vinlinux.org/repos/people/munepi/documents
4                   /10/{name}-{version}.tar.gz
5 BuildRequires:   ncurses-devel
6
7 BuildRoot:       %{_tmppath}/%{name}-{version}-{release}-root
8 ...
9
10 %changelog
11 * Sat May 29 2010 YOUR NAME <YOUR E-MAIL ADDRESS> 0.2-2
12 - added BuildRequires: ncurses-devel
13
14 * Sat May 29 2010 YOUR NAME <YOUR E-MAIL ADDRESS> 0.2-1
```



```
15 - new upstream release
16 ...
```

再び, `rpmbuild -ba` で `echowlv1` パッケージをビルドして下さい. `echowlv1-0.2-2v15.i386.rpm`, `echowlv1-0.2-2v15.src.rpm` が生成されるはずです.

```
1 $ rpmbuild -ba ~/rpm/SPECS/echowlv1-v1.spec
```

今度は, `vbuilder` に `echowlv1-0.2-2v15.src.rpm` を投げてみましょう. `echowlv1-0.2-2v15.i386.rpm` を返してくれたら, パッケージのビルドテストに成功です.

```
1 $ sudo vbuilder --version 5.1 clean \
2   build-rpm ~/rpm/SRPMs/echowlv1-0.2-2v15.src.rpm
```

`vbuilder` が返してくれた `echowlv1-0.2-2v15.i386.rpm` のパッケージをテストして下さい. パッケージのインストール, アンインストールも問題なければ, 入門編は終了です. お疲れ様でした.