RPM パッケージ作成ハンズオン　入門編

山本　宗宏 <munepi@vinelinux.org>

2011 年 7 月 17 日@株式会社グッデイ

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するセミナーです．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/11/lms201107b.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 または Vine Linux 6 がインストールされた計算機を用意して下さい．このレジュメでは，Vine Linux 6 i686 を前提として解説しています．もし 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．

Vine Linux 6

```
# %packager Your Name   <your mail address>
# %packager     Munehiro Yamamoto <munepi@vinelinux.org>
```

```
%packager       munepi

# GnuPG によるパッケージ署名を行う
# %_signature gpg
# %_gpg_name Your Name <your mail address>

# ソースファイル置き場をパッケージ別にする
%_sourcedir %{_topdir}/SOURCES/%{name}
```

Vine Linux 5

```
%_topdir ${HOME}/rpm

# %packager Your Name   <your mail address>
# %packager     Munehiro Yamamoto <munepi@vinelinux.org>
%packager       munepi

# gpg signing
# %_signature gpg
# %_gpg_name Your Name <your mail address>

# more useful macros
%_sourcedir     %{_topdir}/SOURCES/%{name}
```

- %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．デフォルトでは，%_sourcedir %{_topdir}/SOURCES となっています．

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．

Vine Linux 6

あなたはおそらく手元の計算機の管理者です．念のために，id コマンドを実行して，wheel グループに所属しているか確認して下さい．

```
$ id
uid=1000(munepi) gid=1000(munepi) groups=1000(munepi),10(wheel)
```

Vine Linux 5

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．

```
$ 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．

```
$ 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．

```
$ 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．

```
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
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

wget コマンドを用いて，echowlvl-0.1.tar.gz をダウンロードします．

```
$ wget http://trac.vinelinux.org/repos/people/munepi/documents/11/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．

```
$ 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．

```
$ tar xvf echowlvl-0.1.tar.gz
$ cd echowlvl-0.1
$ ./configure
$ make
$ ./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うべき 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，spec ファイルの雛形を用意しました．wget コマンドを用いて，echowlvl-vl.spec をダウンロードして下さい．

```
$ wget http://trac.vinelinux.org/repos/people/munepi/documents/11/echowlvl
    -vl.spec
```

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

```
$ 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．

```
## -*- coding: utf-8-unix -*-
## This is a template spec file for LILO 2011.07 Monthly Seminar.
Name:           echowlvl
Version:        0.1
Release:        1%{?_dist_release}
Summary:        Sample program for RPM making hands-on (lms201107)
Summary(ja):    RPM パッケージ作成ハンズオン (lms201107) 用サンプルプログラム
Group:          Applications/Other
License:        GPLv3+
URL:            http://trac.vinelinux.org/wiki/OfflineMeeting/20110711
Source0:        http://trac.vinelinux.org/repos/people/munepi/documents
    /11/%{name}-%{version}.tar.gz

BuildRoot:      %{_tmppath}/%{name}-%{version}-%{release}-root
```

```
14
15 Distribution:   YOUR LINUX DISTRIBUTION
16 Vendor:         YOUR VENDOR
17 Packager:       YOUR NAME (or YOUR ID)
18
19 %description
20 echowlvl is a sample program for RPM making hands-on (lms201107).
21
22 %description -l ja
23 echowlvlは RPM パッケージ作成ハンズオン (lms201107) 用サンプルプログラムで
       す.
```

- 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 としてください．
- Release：リリース番号と Vine Linux のバージョンを組み合せて書きます．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．例えば，Vine Linux 6 の場合は，1%{?_dist_release} は 1vl6 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アーカイブを展開し，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．

```
%files
%defattr(-,root,root,-)
```

さて，何を書けばよいのでしょう？ 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．

```
%changelog
* Sun Jul 17 2011 YOUR NAME <YOUR E-MAIL ADDRESS> 0.1-1
- first release

## 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．

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

では，rpmbuild -ba でパッケージをビルドしてみます．

```
$ rpmbuild -ba ~/rpm/SPECS/echowlvl-vl.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．

```
%files
%defattr(-,root,root,-)
%doc AUTHORS COPYING ChangeLog INSTALL NEWS README
%{_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．

```
$ sudo apt-get install ~/rpm/RPMS/i686/echowlvl-0.1-1vl6.i686.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るでしょうか？

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．

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

echowlvl コマンドを実行できてしまったりしないか，確認してみましょう．

```
$ echowlvl
bash: echowlvl: コマンドが見つかりません
$ which echowlvl
/usr/bin/which: no echowlvl in (/home/munepi/bin:/usr/local/bin:/usr/bin:/
    bin:/usr/local/sbin:/usr/sbin:/sbin)
```

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！

wget コマンドを用いて，echowlvl-0.2.tar.gz をダウンロードして下さい．

```
$ wget http://trac.vinelinux.org/repos/people/munepi/documents/11/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．

3.1.1 ビルドに必要なライブラリを追加

もし echowlvl をコンパイルできなかった方は，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だったようです．/usr/include/curses.h は，ncurces-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．

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

3.1.2 spec ファイルを更新

echowlvl の新しいバージョンに合わせて，echowlvl-vl.spec を編集しましょう．`Version` はバージョンを `0.2` にして下さい．`%changelog` にソフトウェアが更新されたことを記します．

```
## -*- coding: utf-8-unix -*-
## This is a template spec file for LILO 2011.07 Monthly Seminar.
Name:           echowlvl
Version:        0.2
Release:        1%{?_dist_release}
...

%changelog
```

```
* Sun Jul 17 2011 YOUR NAME <YOUR E-MAIL ADDRESS> 0.2-1
- new upstream release

* Sun Jul 17 2011 YOUR NAME <YOUR E-MAIL ADDRESS> 0.1-1
- first release

## end of file
```

3.1.3 パッケージを更新

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！

4 ビルドテストしましょう

パッケージができたら，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よね．「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ひんよ？」，「私の環境でもビルドできたけど，この機能が使えへんよ？」といったことが起こると不都合ですよね．

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 の基本操作

vbuilder は vbootstrap パッケージに格納されています．

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

基本的な流れは以下の通りです．

1. 必要があれば，過去に構築した子環境を削除

   ```
   $ sudo vbuilder clean
   ```

2. 子環境に最小環境を構築

   ```
   $ sudo vbuilder build
   ```

3. 子環境上で src.rpm をビルド

   ```
   $ sudo vbuilder build-rpm hoge-1.0-1vl6.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．

```
$ sudo vbuilder clean build build-rpm hoge-1.0-1vl6.src.rpm
```

この場合，build を省略できます．

```
$ sudo vbuilder clean build-rpm hoge-1.0-1vl6.src.rpm
```

なお，$ sudo vbuilder clean build を実行すると，デフォルトで/var/local/vbootstrap/6.0_i386 が約 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-0.2-1vl6.i686.rpm といっしょに，もう 1 つ echowlvl-0.2-1vl6.src.rpm という rpm ができていました．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-0.2-1vl6.src.rpm から echowlvl-0.2-1vl6.i686.rpm をビルドできます．

vbuilder に echowlvl-0.2-1vl6.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-0.2-1vl6.i686.rpm を返してくれるでしょうか？

```
$ sudo vbuilder clean build-rpm ~/rpm/SRPMS/echowlvl-0.2-1vl6.src.rpm
```

4.3 依存関係を解決する

vbuilder に echowlvl-0.2-1vl6.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-0.2-1vl6.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-0.2-1vl6.src.rpm がビルドできそうです．

早速，試してみましょう．さきほど vbuilder で作った子環境上に ncurses-devel パッケージをインストールします．

```
$ sudo vbuilder install-rpm ncurses-devel
```

再び，vbuilder に echowlvl-0.2-1vl6.src.rpm を投げてみます．echowlvl-0.2-1vl6.i686.rpm を返してくれるでしょうか？

```
$ sudo vbuilder build-rpm ~/rpm/SRPMS/echowlvl-0.2-1vl6.src.rpm
```

4.3.1 spec ファイルを更新

echowlvl-vl.spec を編集しましょう．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．

```
## -*- coding: utf-8-unix -*-
## This is a template spec file for LILO 2011.07 Monthly Seminar.
Name:           echowlvl
```

```
Version:        0.2
Release:        2%{?_dist_release}
...
Source0:        http://trac.vinelinux.org/repos/people/munepi/documents
    /11/%{name}-%{version}.tar.gz

BuildRequires:  ncurses-devel

BuildRoot:      %{_tmppath}/%{name}-%{version}-%{release}-root
...

%changelog
* Sun Jul 17 2011 YOUR NAME <YOUR E-MAIL ADDRESS> 0.2-2
- added BuildRequires: ncurses-devel

* Sun Jul 17 2011 YOUR NAME <YOUR E-MAIL ADDRESS> 0.2-1
- new upstream release

* Sun Jul 17 2011 YOUR NAME <YOUR E-MAIL ADDRESS> 0.1-1
- first release

## end of file
```

4.3.2 パッケージのビルドテスト

再び，rpmbuild -ba で echowlvl パッケージをビルドして下さい．echowlvl-0.2-2vl6.i686.rpm, echowlvl-0.2-2vl6.src.rpm が生成されるはずです．

今度は，vbuilder に echowlvl-0.2-2vl6.src.rpm を投げてみましょう．一旦，vbuilder clean で小環境を削除します．vbuilder build-rpm で小環境を再構築し，echowlvl-0.2-2vl6.src.rpm をビルドします．echowlvl-0.2-2vl6.i686.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．

```
$ sudo vbuilder clean build-rpm ~/rpm/SRPMS/echowlvl-0.2-2vl6.src.rpm
```

vbuilder が返してくれた echowlvl-0.2-2vl6.i686.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，動作確認，アンインストールに問題なければ，入門編は終了です．お疲れ様でした．