

はじめに

この文書は [NetBSD Advent Calendar 2017](#) の 19 日目の記事として書かれました。

pkgsrc でパッケージをアップデートする時に、[pkg_rolling-replace を使ってみよー](#)という記事です。(そのまんま。)

なお、以下に書かれていることは全て pkg_rolling-replace の man に記載があります。この記事は、試してみた、という以上の内容はありません。さらに詳しい情報が知りたい場合は、man 読んで下さい。

pkg_rolling-replace って？

最近、バイナリーパッケージを pkgin で使っている人も増えてきたかもしれません。しかし、pkgin に慣れなかったり、公式のバイナリーパッケージに適用されている各種オプションが自分と合わなかったり、昔からずっと pkgsrc を使っている等の理由で、pkgsrc を使っている人もまだまだ多いと思います。

pkg_rolling-replace は、pkgsrc/pkgtools/pkg_rolling-replace にあるツールです。2006 年 12 月に登場しました。commit log には

```
pkg_rolling-replace does 'make replace' on one package at a time,
tsorting the packages being replaced according to their
interdependencies, which avoids most duplicate rebuilds.
```

とあります。

似たようなアイデアは以前からあったのでしょうけれども、2005 年 5 月に tech-pkg メーリングリストで流れた『[Smarter make update / pkg_chk algo](#)』などが良い例だと思います。(これに投稿されたスクリプトを少しだけいじって、使ってみたりしていました。)

pkg_rolling-replace が登場してからは、アップデートはこれに任せることが多くなりました。

使い方

前提

pkgsrc を使える環境であることです。pkgsrc 自体をアップデートしていないと、パッケージもアップデートされないので、注意が必要です。

あるパッケージをアップデートしたい

大抵の場合は、特定のパッケージをアップデートしたいんだけど、make update しちゃうとひどい目にあうので、対象のパッケージと、依存関係を持つパッケージだけ、make replace したくなります。それができます。

まず、アップデートしたいパッケージにマークをつけます。今回は、security/ruby-oauth パッケージを対象にすることにします。パッケージ名を確認します。

```
% pkg_info | grep oauth
```

ruby23-oauth-0.5.1 OAuth Core Ruby implementation

ruby23-oauth だとわかったので、これをマークします。(複数あるときは、複数マークして下さい。)

```
pkg_admin set rebuild=YES ruby23-oauth
```

もし、間違えてマークしまった時は、

```
pkg_admin set rebuild='' ruby23-oauth
```

とすればよいです。

あとは、pkg_rolling-replace を実行するだけです。

```
pkg_rolling-replace -rsv
```

-r をつけると、バイナリーパッケージを作成して保存しません。自動的に行われるアップデート作業の中で、バイナリーパッケージは作成されているので、それをとっておきたい場合は、-r をはずして下さい。

-s は strict mode を有効にします。これをつけていないと、依存関係をもつソフトウェアの更新がされず、結果的に更新対象のソフトが更新後動作しないことがあるので、悩みたくない人は、有効にしておいた方がよいです。

-v はおさっしのとおり、コマンドの作業途中の出力がより冗長になります。自分はどこまで作業が進んだかみたいので、いつもつけています。

全てのパッケージをアップデートしたい

ひとつだけ、じゃなくて、全てのパッケージをアップデートしたい時は、-u オプションをつけます。つまり、

```
pkg_rolling-replace -rsuv
```

で、現在インストールしているパッケージが、全てアップデートされます。

QA

途中で止まっちゃうんだけど

たとえば、以下のような感じで止まっちゃうとします。

```
<snip>
=> Checking for work-directory references in ruby23-base-2.3.6
=> Creating binary package /usr/pkgsrc/lang/ruby23-base/work.x86_64/.packages/ruby23-base-2.3.6.tgz
====> Building binary package for ruby23-base-2.3.6
=> Creating binary package /usr/packages/x86_64/All/ruby23-base-2.3.6.tgz
====> Installing binary package of ruby23-base-2.3.6
pkg_add: A different version of ruby23-base-2.3.6 is already installed: ruby23-base-2.3.5
pkg_add: 1 package addition failed
*** Error code 1
```

```
Stop.  
make[3]: stopped in /usr/pkgsrc/lang/ruby23-base  
*** Error code 1  
  
Stop.  
make[2]: stopped in /usr/pkgsrc/lang/ruby23-base  
*** Error code 1  
  
Stop.  
make[1]: stopped in /usr/pkgsrc/lang/ruby23-base  
*** Error code 1  
  
Stop.  
make: stopped in /usr/pkgsrc/security/ruby-oauth  
*** 'make replace' failed for package ruby23-oauth.  
*** Please read the errors listed above, fix the problem,  
*** then re-run pkg_rolling-replace to continue.  
- ruby23-oauth
```

こうしたときは、既に ruby23-base パッケージができていますので、できているパッケージで replace してしまいます。

```
pkg_add -U /usr/pkgsrc/lang/ruby23-base/work.x86_64/.packages/ruby23-base-2.3.6.tgz
```

そして、pkg_rolling-replace を再実行すると、続きが走るの、よいと思います。

ただし、途中で止まっちゃったパッケージがそもそも make install できなくなっちゃっているパッケージだと、手も足もでません。(時々そういうのがあったりします。)

dry run したい

-n オプションで可能です。つまり、

```
pkg_rolling-replace -nrsv
```

などでできます。

サンプル

ruby23-oauth パッケージをアップデートした結果は、以下のとおりです。ruby23-oauth -> ruby23-base -> ruby23-mikutter とアップデートが行われています。(ruby23-base は手動で pkg_add していますけれども。)

あとがき

NetBSD 使っているひとつ、みんな古くから使っていて初心者なんかいなさそうで(そんなことはない) 英語なんかもすらすら読めて(ぼくはちがう) デバイスドライバ書いたりカーネルいじったりしていて(ぼくは超ちがう) っていうすごいはっかー、という感じがします。

そんなことは全然全くさっぱりちっともないただの利用者たるぼくですが、何か書いてみることにしました。誰かのお役に立てれば幸いです。

ではでは。

明日の Advent Calendar は

610t さんの『[NanoPi Neo2 で NetBSD を動かしてみた](#)』です。楽しみです。