

はじめに

この文書は [NetBSD Advent Calendar 2016](#) の 22 日目の記事として書かれました。(その後、少しアップデートされています。)

`pkgsrc` は、NetBSD の公式パッケージシステムです。このパッケージシステムの特徴として、他の UN*X 系 OS でも使うことができます。

他の *BSD でいう、"ports" にあたるものです。(ちなみに他の *BSD でいう "packages" は、NetBSD では、"binary packages" という言い方をします。)

使い始める時や、その後のアップデートにちょっとだけ便利な Makefile を紹介します。

pkgsrc はどうやって使い始めて、その後最新を保っていけばいいの？

公式の方法

[公式の pkgsrc guide](#) では、`cvs` を使うやり方が紹介されています。`cvs` は、NetBSD では base system に入っているの、いきなり、`cvs` で取得してもよいですし、`ftp` or `http` で snapshot を取ってきて、そのあと、`cvs` で最新にするのもいいです。後者の方法だとこんな感じです。

```
# cd /tmp
# ftp https://cdn.netbsd.org/pub/pkgsrc/current/pkgsrc.tar.xz
# cd /usr
# tar xzf /tmp/pkgsrc.tar.xz
# cd pkgsrc
# cvs update -dP
```

[cdn.netbsd.org](#) は、今年の 7 月にできた新しいミラーで、高速です。https にも対応しています。

さて、`cvs update -dP` はどれくらいかかるでしょうか。[ConoHa](#) 上の NetBSD サーバで 2016/12/22 に試してみたところ、151 秒かかりました。disk が速いととても速いですが、disk が遅いとかなりかかります。

`cvs update -dP` が終わり、最新にアップデートした状態で、すぐ 2 回めを行ってみます。全くアップデートがないはずですが、同じくらいかかります。(141 秒かかりました。)もし、アップデートがあれば、さらにかかります。要するに遅いのです。

また、NetBSD 以外で `pkgsrc` を使おうとすると、たいてい(?) `cvs` をインストールしないと行けません。ちょっとめんどろ。

git を使う方法

公式の方法ではないのですが、`git` を使う方法があります。よく知られているのは、NetBSD の開発者の方が GitHub にミラーされているリポジトリを使う方法です。snapshot はないので、いきなり clone します。NetBSD 公式の GitHub ミラーが誕生したので、そこから clone します。(なお、zip ダウンロード後、それを展開した方が、低速回線では早いです。)

```
# cd /usr
# git clone --depth 1 git://github.com/NetBSD/pkgsrc.git
```

初回にかかる時間はネットワークの速さにかなり依存します。cvs と同じく、ConoHa 上の NetBSD サーバで 2016/12/22 に試してみたところ、95 秒かかりました。snapshot をとってきたあとに、cvs をつかってアップデートする方法よりも速かったです。

さて、cvs のときと同様に、最新にアップデートした状態で、アップデート (git pull) をすぐ行ってみます。

```
# cd /usr/pkgsrc
# git pull
```

全くアップデートがないので、すぐ終わります。1 秒かかりませんでした。アップデートがあっても cvs と比べてとても速いです。

また、NetBSD 以外で pkgsrc を使おうとする場合、最近はたいてい(?) git は入っているような気がします。そういう場合は楽です。

また、pkgsrc-wip も使おうとする場合は、git を使う方法しかありません。どちらも git の方がわかりやすいかもしれません。

/usr/Makefile

DragonFly BSD の Makefile

git を使うにしても、新規に利用を開始する際、記憶力の弱いわたしは、いつも git コマンドの引数を忘れてしまいます。また、pkgsrc-wip はどうだったとか、NetBSD 上で使う場合、src や xsrc も git で取ってきたくくなります。引数はどうだったっけ？

そこで、思い出したのが、DragonFly BSD でした。DragonFly BSD は一時期、公式パッケージシステムとして、pkgsrc を使っており、DragonFly BSD プロジェクト独自で、git リポジトリに pkgsrc をインポートして、それを使うようになっていました。

便利だったのは、/usr/Makefile が用意されており、そこに使い始めとアップデートのためのターゲットが書かれていたことです。具体的には、

```
# cd /usr
# make
```

と入力すると、ヘルプとしてターゲットの一覧が出てきて、あとは選ぶだけでした。

```
# make pkgsrc-create-shallow
```

これだけで、あとは使い始められました。アップデートも、

```
# cd /usr
# make pkgsrc-update
```

とするだけ。引数がどうだったとか考えなくてもよく、便利でした。

NetBSD 用を作ってみる

この、DragonFly BSD の /usr/Makefile を真似て、/usr/Makefile を作ってみました。

NetBSD 用 (BSD make 用): <http://www.rururu.org/~isihara/Makefile.usr.bmake>

NetBSD 用のは、pkgsrc に加え、pkgsrc-wip・src・xsrc 用のターゲットがあり、使うことができます。こんな感じ。

```
# cd /usr
# ftp -o Makefile http://www.rururu.org/ isihara/Makefile.usr.bmake
# make
HELP:
  make pkgsrc-create           - fetch initial pkgsrc repository from the net
  make pkgsrc-create-shallow  - fetch initial history-free pkgsrc repository
  make pkgsrc-create-repo     - fetch pkgsrc repository only, no checkout

  make pkgsrc-wip-create      - fetch initial pkgsrc-wip repository from the net
  make pkgsrc-wip-create-shallow - fetch initial history-free pkgsrc-wip repository
  make pkgsrc-wip-create-repo - fetch pkgsrc-wip repository only, no checkout

  make src-create             - fetch initial src repository from the net
  make src-create-shallow    - fetch initial history-free src repository
  make src-create-repo       - fetch src repository only, no checkout

  make xsrc-create            - fetch initial xsrc repository from the net
  make xsrc-create-shallow   - fetch initial history-free xsrc repository
  make xsrc-create-repo      - fetch xsrc repository only, no checkout
```

あとは、make なんとかするだけ。たとえば、

```
# make pkgsrc-create-shallow
# make pkgsrc-wip-create-shallow
```

とすると、/usr/pkgsrc と、/usr/pkgsrc/wip ができます。

アップデートもかんたん。ターゲットを覚えておく必要はなく、

```
# cd /usr
# make
HELP:
  make pkgsrc-update          - update your pkgsrc repo from the net

  make pkgsrc-wip-update     - update your pkgsrc-wip repo from the net

  make src-create            - fetch initial src repository from the net
  make src-create-shallow   - fetch initial history-free src repository
  make src-create-repo      - fetch src repository only, no checkout

  make xsrc-create           - fetch initial xsrc repository from the net
  make xsrc-create-shallow  - fetch initial history-free xsrc repository
  make xsrc-create-repo     - fetch xsrc repository only, no checkout
```

と、必要なターゲットはヘルプを見ればよいのです。

NetBSD 以外でも使いたい

自分が CentOS 上でも pkgsrc を使っており、NetBSD 上と同様に /usr/Makefile が同様に欲しくなり、GNU make 用の /usr/Makefile もみようみまねで作ってみました。

NetBSD 以外用 (GNU make 用): <http://www.rururu.org/~isihara/Makefile.usr.gmake>

こちらは、pkgsrc と pkgsrc-wip のみです。全く同様に使うことができます。

```
# cd /usr
# curl http://www.rururu.org/ isihara/Makefile.usr.gmake > Makefile
# make
HELP:
  make pkgsrc-create           - fetch initial pkgsrc repository from the net
  make pkgsrc-create-shallow   - fetch initial history-free pkgsrc repository
  make pkgsrc-create-repo      - fetch pkgsrc repository only, no checkout

  make pkgsrc-wip-create       - fetch initial pkgsrc-wip repository from the net
  make pkgsrc-wip-create-shallow - fetch initial history-free pkgsrc-wip repository
  make pkgsrc-wip-create-repo   - fetch pkgsrc-wip repository only, no checkout
```

QA

取ってくる branch は？ どうやって変えたらいいの？

NetBSD 用のデフォルト設定は以下のとおり。

- pkgsrc
 - current
- pkgsrc-wip
 - master
- src
 - netbsd-7
- xsrc
 - netbsd-7

NetBSD 以外用のデフォルト設定は以下のとおり。

- pkgsrc
 - current
- pkgsrc-wip
 - master

変更のしかたは、create する前に、Makefile を直接編集してもいいし、

```
# make pkgsrc-create-shallow BRANCHPS=pkgsrc-2016Q3
```

などと create する時に、make の引数にしてもいいよ。(Makefile の先頭に BRANCH なんとか、という変数が定義してあるので、その変数名で指定します。)

pkgsrc の 4 半期リリースの切り替えはどうやったらいいの？

DragonFly BSD の(当時の)公式ドキュメントのとおりで、コピペになるんだけど、以下のとおり。

例として、2016Q1 から、2016Q2 に切り替える場合を示してみる。

```
# cd /usr/pkgsrc
# git branch pkgsrc-2016Q2 origin/pkgsrc-2016Q2
Branch pkgsrc-2016Q2 set up to track remote branch pkgsrc-2016Q2 from origin.
# git checkout pkgsrc-2016Q2
Checking out files: 100% (11006/11006), done.
Switched to branch 'pkgsrc-2016Q2'
# git pull
remote: Counting objects: 9, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 9 (delta 6), reused 9 (delta 6), pack-reused 0
```

```
Unpacking objects: 100% (9/9), done.  
From git://github.com/NetBSD/pkgsrc  
 0d1dde8..3ce9352 trunk -> origin/trunk  
Already up-to-date.  
# git branch  
  pkgsrc-2016Q1  
* pkgsrc-2016Q2
```

あとは、今までどおり、

```
# cd /usr  
# make pkgsrc-update
```

で、2016Q2 をアップデートできます。

NetBSD 以外用が使えないよ

CentOS6 の GNU make (GNU Make 3.81) 以外ではテストしてないので、だめかもしれない。もしそうであれば、ごめんなさい。(どこをどうやって直したらいいか、教えてあげるぞ、という親切な方は、@tishara に連絡くれるとうれしいです。)

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、ていう感じがします。

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。

もし、あなたが pkgsrc の情報を探しているのであれば、"[pkgsrc めも](#)" もご覧いただくと、ほしい情報があるかもしれません。

ではでは。

明日の Advent Calendar は

ひきつづきぼくです。何書くかまだ決まっていません(汗 "[NetBSD でつかえる USB 無線 LAN アダプターの調べ方](#)" という文書を書きました。