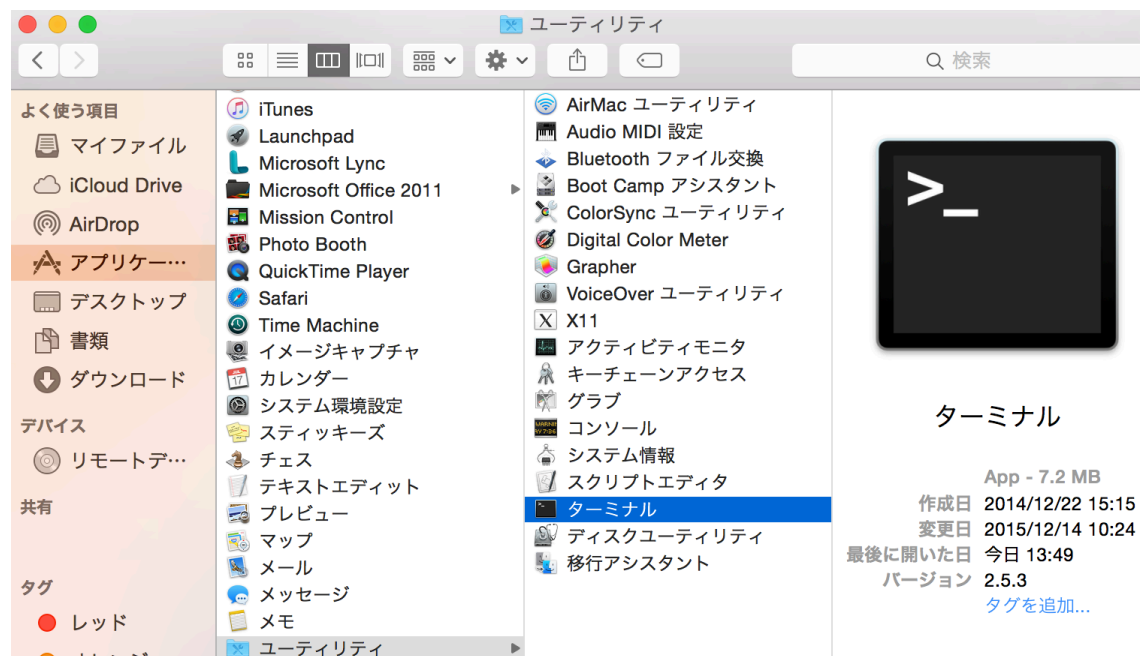


# UNIX環境の構築 (OSX編)

Mac環境でUNIXコマンドを使用するには



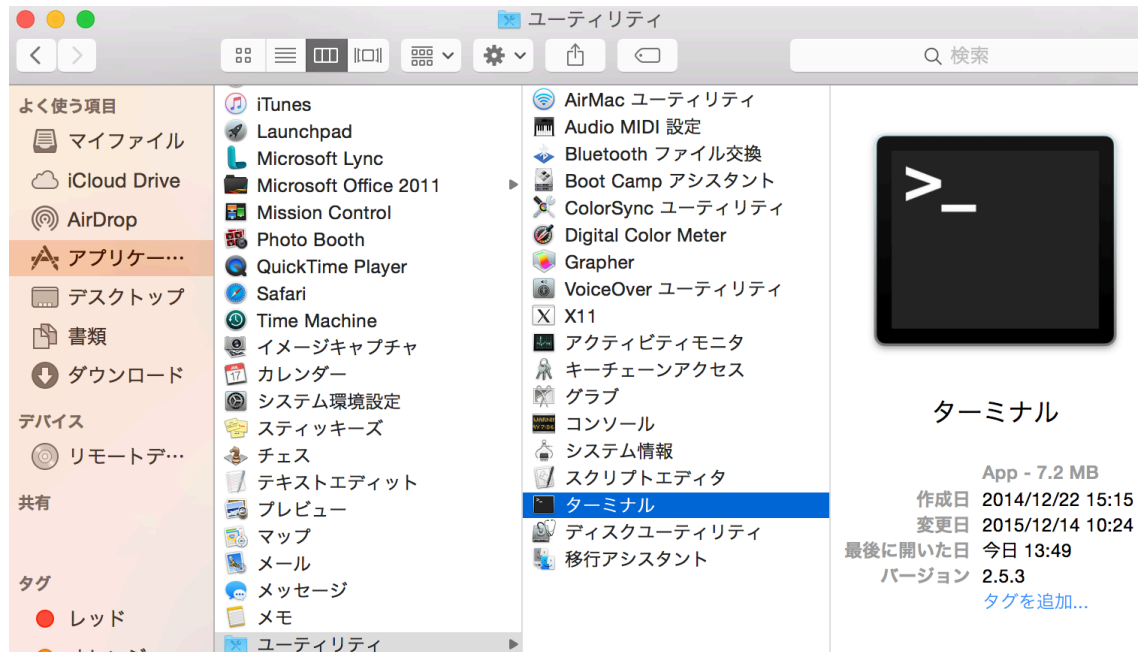
# 概要

- OSXは元々Unix系OSであるDarwin上で動作している。そのため、ターミナル.app から標準的なUnixコマンドを使用することが可能である。
- しかし、本格的にUnixコマンドを使用した解析操作を行おうとすると、必要なソフトウェアが動作しない／インストールされていないため、つまづくことになる。
- 本資料はOSXにおいて必要なソフトウェアをインストールし、解析操作に必要なUnix環境を構築する方法を取り上げる。

# ターミナル.app



➤ [アプリケーション] -> [ユーティリティ] の中に、ターミナル.app がある。実行することで、シェルが起動する。



# 方法の概要

- 1)  
Xcode の Command Line Tools をインストールする。
- 2)  
パッケージシステムをインストールする。  
(Macports, Homebrew, etc)
- 3)  
パッケージシステムを用いて、ソフトウェアをインストールする。

# Xcode



## ➤ Xcode

( <https://developer.apple.com/jp/xcode/index.html> )

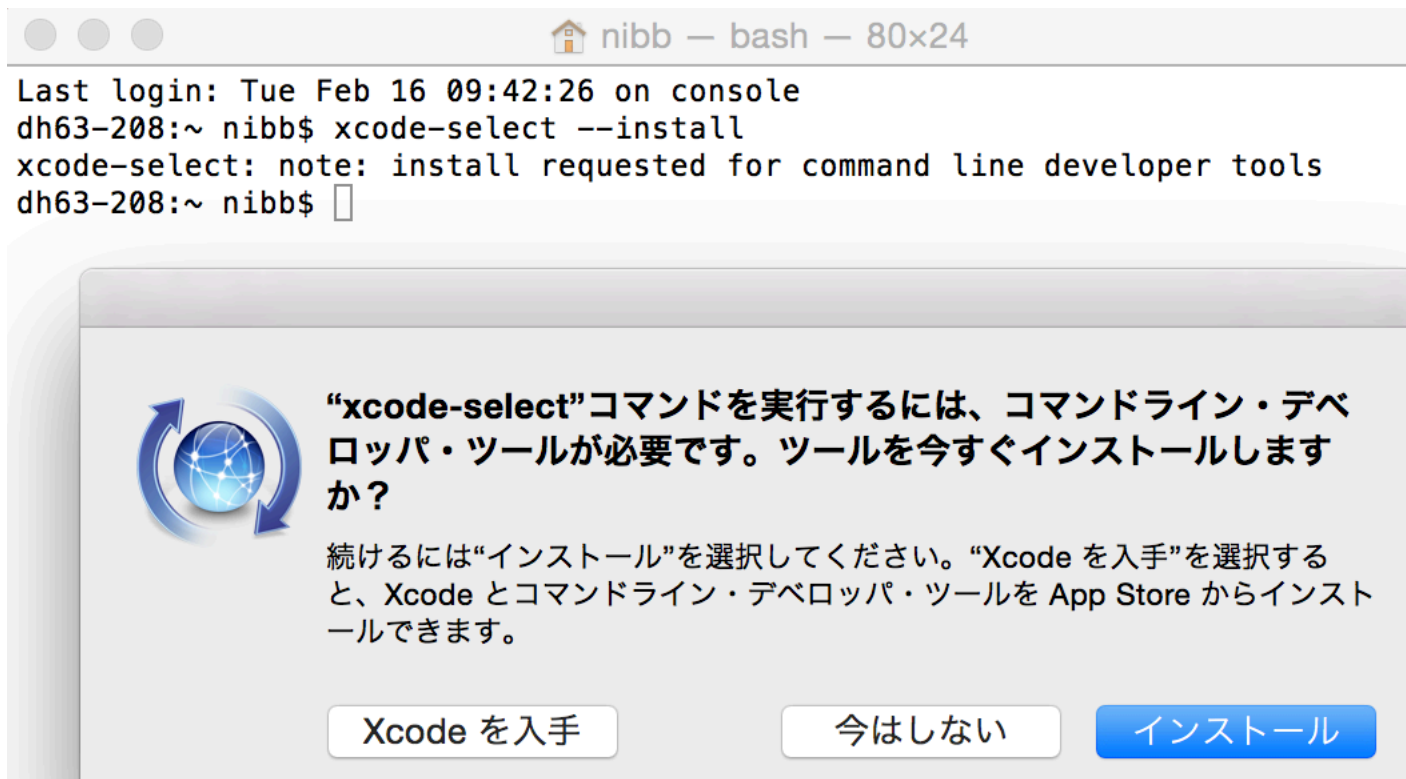
- Xcode はソフトウェアを開発するための統合環境である。米 Apple 社から OSX用に無償で配布されている。
- ダウンロードは App Store から行う。  
(容量が 4 G超と巨大なので注意)

# Command Line Tools

- 前述のXcodeは、統合環境であり色々なものが同梱されている。その中で、ソースファイルからコンパイルをするために必要なのは、Xcodeに付随する Command Line Tools である。
- Commande Line Tools のみインストールしたい場合、ターミナル.app から以下のコマンドを入力することで、インストールを行うことができる。

```
$ xcode-select --install
```

# Command Line Tools



# (例) make

```
dh63-208:Desktop nibb$ cd samtools-1.3
dh63-208:samtools-1.3 nibb$ make
Makefile:99: config.mk: No such file or directory
gcc -g -Wall -O2 -D_FILE_OFFSET_BITS=64 -D_LARGEFILE64_SOURCE -I. -Ihtslib-1.3
-c -o bam_index.o bam_index.c
gcc -g -Wall -O2 -D_FILE_OFFSET_BITS=64 -D_LARGEFILE64_SOURCE -I. -Ihtslib-1.3
-c -o bam_plcmd.o bam_plcmd.c
gcc -g -Wall -O2 -D_FILE_OFFSET_BITS=64 -D_LARGEFILE64_SOURCE -I. -Ihtslib-1.3
-c -o sam_view.o sam_view.c
gcc -g -Wall -O2 -D_FILE_OFFSET_BITS=64 -D_LARGEFILE64_SOURCE -I. -Ihtslib-1.3
-c -o bam_cat.o bam_cat.c
gcc -g -Wall -O2 -D_FILE_OFFSET_BITS=64 -D_LARGEFILE64_SOURCE -I. -Ihtslib-1.3
-c -o bam_md.o bam_md.c
gcc -g -Wall -O2 -D_FILE_OFFSET_BITS=64 -D_LARGEFILE64_SOURCE -I. -Ihtslib-1.3
-c -o bam_reheader.o bam_reheader.c
gcc -g -Wall -O2 -D_FILE_OFFSET_BITS=64 -D_LARGEFILE64_SOURCE -I. -Ihtslib-1.3
-c -o bam_sort.o bam_sort.c
gcc -g -Wall -O2 -D_FILE_OFFSET_BITS=64 -D_LARGEFILE64_SOURCE -I. -Ihtslib-1.3
-c -o bedidx.o bedidx.c
bedidx.c:41:1: warning: unused function 'ks_getc' [-Wunused-function]
KSTREAM_INIT(gzFile, gzread, 8192)
^
htslib-1.3/htslib/kseq.h:154:49: note: expanded from macro 'KSTREAM_INIT'
#define KSTREAM_INIT(type_t, __read, __bufsize) KSTREAM_INIT2(static, ty...
```

一例として、samtools をmakeしてみた結果。最後にはきちんと作成された。



# パッケージシステム

- Unix環境にソフトウェアをインストールする場合、ソフトウェアに関連付けされているファイル一式をまとめた「パッケージ」を用いることが多い。
- それらパッケージを用いて、必要なファイルと依存関係も含めてインストールの補助を行ってくれるのが「パッケージシステム」である。
- OSXで利用できるパッケージシステムは、Macports や Homebrew などが存在する。
- この資料では、Homebrew を使用したソフトウェアのインストールを紹介する。

# Homebrew

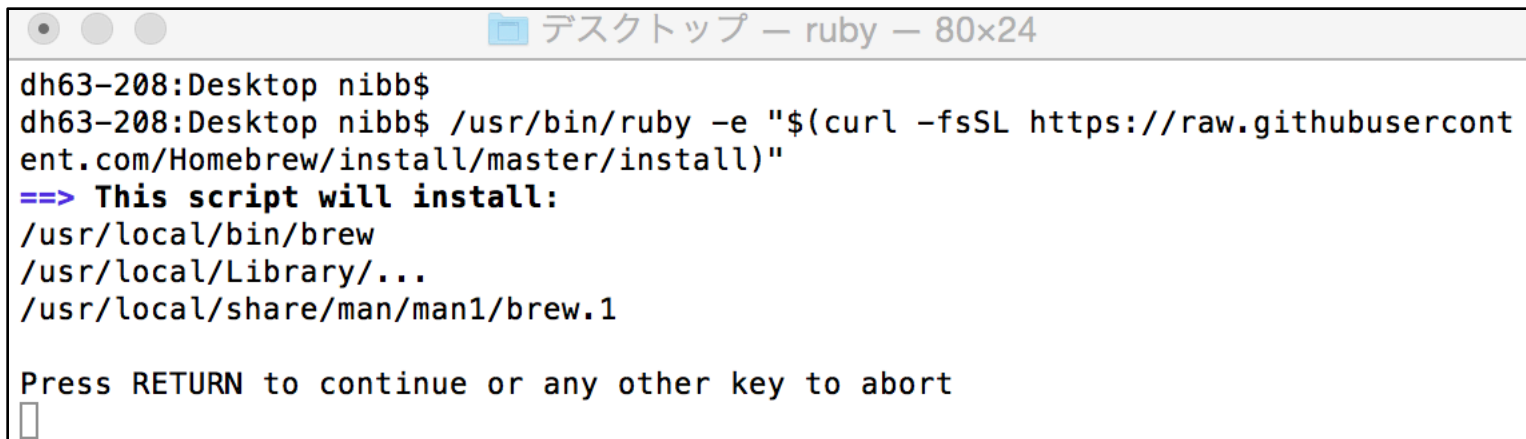
- Homebrew ( [http://brew.sh/index\\_ja.html](http://brew.sh/index_ja.html) )
- Mac OS X上でソフトウェアの導入を単純化するパッケージ管理システムのひとつ。
- 管理者権限がない一般ユーザでも使用が可能。
- パッケージインストール先は /usr/local
- インストールにかかる時間が比較的少ない。

# Homebrew インストール

ターミナル.app を開き、以下のコマンドを入力する。

（長いので、[http://brew.sh/index\\_ja.html](http://brew.sh/index_ja.html) からコピー&ペーストしたほうが良い）

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

A screenshot of a macOS terminal window titled 'デスクトップ - ruby - 80x24'. The terminal shows the execution of the Homebrew installation command. The prompt is 'dh63-208:Desktop nibb\$'. The command entered is '/usr/bin/ruby -e "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"'. The output shows the script's purpose and the paths it will install to: '/usr/local/bin/brew', '/usr/local/Library/...', and '/usr/local/share/man/man1/brew.1'. It then prompts the user to 'Press RETURN to continue or any other key to abort' with a cursor on the next line.

```
dh63-208:Desktop nibb$  
dh63-208:Desktop nibb$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"  
==> This script will install:  
/usr/local/bin/brew  
/usr/local/Library/...  
/usr/local/share/man/man1/brew.1  
  
Press RETURN to continue or any other key to abort  
█
```

リターンキーを押してHomebrewのインストールを続行します。

# Homebrew インストール

Press RETURN to continue or any other key to abort

```
==> /usr/bin/sudo /bin/mkdir /usr/local
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

Password:●

OSX上での管理者権限を持っている場合、パスワードを入力します。  
そうでないと、Homebrew自体のインストールができません。

# Homebrew インストール

```
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

Password:
==> /usr/bin/sudo /bin/chmod g+rxw /usr/local
==> /usr/bin/sudo /usr/sbin/chown nibb:admin /usr/local
==> /usr/bin/sudo /bin/mkdir /Library/Caches/Homebrew
==> /usr/bin/sudo /bin/chmod g+rxw /Library/Caches/Homebrew
==> /usr/bin/sudo /usr/sbin/chown nibb /Library/Caches/Homebrew
==> Downloading and installing Homebrew...
remote: Counting objects: 4044, done.
remote: Compressing objects: 100% (3892/3892), done.
remote: Total 4044 (delta 35), reused 2207 (delta 21), pack-reused 0
Receiving objects: 100% (4044/4044), 3.33 MiB | 3.41 MiB/s, done.
Resolving deltas: 100% (35/35), done.
From https://github.com/Homebrew/homebrew
 * [new branch]      master      -> origin/master
HEAD is now at 53d05af ffms2: prefer berkeley mirror
==> Installation successful!
==> Next steps
Run `brew help` to get started
dh63-208:Desktop nibb$
```

Homebrew のインストールが完了しました。

メッセージに従い、`brew help` と入力すると、Homebrewのヘルプが表示されます。

# Homebrew の基本操作

```
$ brew search パッケージ名
```

示されるパッケージを検索します。

```
$ brew install パッケージ名
```

示されるパッケージをインストールします。

```
$ brew uninstall パッケージ名
```

示されるパッケージをアンインストールします。

```
$ brew list
```

インストールされたパッケージ一覧を表示します。

# Homebrew の使用例

Homebrew を使って、bowtie2 をインストールする例を紹介します。  
(参照 : <http://bowtie-bio.sourceforge.net/bowtie2/index.shtml> )

```
dh63-208:Desktop nibb$ brew search bowtie2
homebrew/science/bowtie2
```

brew search で、bowtie2を探します。

```
dh63-208:Desktop nibb$ brew search bowtie2
homebrew/science/bowtie2
dh63-208:Desktop nibb$ brew install bowtie2
Error: No available formula with the name "bowtie2"
==> Searching for similarly named formulae...
Error: No similarly named formulae found.
==> Searching taps...
This formula was found in a tap:
homebrew/science/bowtie2
To install it, run:
  brew install homebrew/science/bowtie2
```

brew install で、bowtie2をインストールしようと試みましたが、エラーが出てしまいました。指摘されているように、代わりに brew install homebrew/science/bowtie2 と入力しましょう。

# Homebrew の使用例

```
dh63-208:~ nibb$ brew install homebrew/science/bowtie2
==> Tapping homebrew/science
Cloning into '/usr/local/Library/Taps/homebrew/homebrew-science'...
remote: Counting objects: 580, done.
remote: Compressing objects: 100% (579/579), done.
remote: Total 580 (delta 2), reused 75 (delta 0), pack-reused 0
Receiving objects: 100% (580/580), 481.41 KiB | 458.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.
Checking connectivity... done.
Tapped 573 formulae (601 files, 1.5M)
==> Installing bowtie2 from homebrew/science
==> Installing dependencies for homebrew/science/bowtie2: tbb
==> Installing homebrew/science/bowtie2 dependency: tbb
==> Downloading https://homebrew.bintray.com/bottles/tbb-4.4-20151115.yosemite.b
##### 100.0%
==> Pouring tbb-4.4-20151115.yosemite.bottle.tar.gz
🍺 /usr/local/Cellar/tbb/4.4-20151115: 108 files, 1.9M
==> Installing homebrew/science/bowtie2
==> Downloading https://homebrew.bintray.com/bottles-science/bowtie2-2.2.6.yosem
##### 100.0%
==> Pouring bowtie2-2.2.6.yosemite.bottle.3.tar.gz
🍺 /usr/local/Cellar/bowtie2/2.2.6: 76 files, 20.4M
```

うまくインストールできたようです。 Bowtie2を実行してみましょう。



# Homebrew の使用例

```
dh63-208:~ nibb$ bowtie2
No index, query, or output file specified!
Bowtie 2 version 2.2.6 by Ben Langmead (langmea@cs.jhu.edu, www.cs.jhu.edu/~langmea)
Usage:
  bowtie2 [options]* -x <bt2-idx> {-1 <m1> -2 <m2> | -U <r>} [-S <sam>]

<bt2-idx>  Index filename prefix (minus trailing .X.bt2).
           NOTE: Bowtie 1 and Bowtie 2 indexes are not compatible.
<m1>      Files with #1 mates, paired with files in <m2>.
           Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
<m2>      Files with #2 mates, paired with files in <m1>.
           Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
<r>       Files with unpaired reads.
           Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
<sam>     File for SAM output (default: stdout)
```

オプション無しで実行したので、説明文が出てきました。  
バージョンは 2.2.6 となっています。

これでbowtie2 のインストールが完了していることを確認できました。

# (補) OSX 10.11 での注意点

OSX 10.11 (El Capitan) では、ユーザー権限周りでの変更が行われている。  
/usr, /sbin を始めとするディレクトリの操作が、デフォルトではできない。

参考)

<http://rcmdnk.github.io/blog/2015/10/10/computer-mac/>

```
$ less /System/Library/Sandbox/rootless.conf
```

このため、OSX 10.11 をクリーンインストールした場合、管理者権限を持つ  
ユーザーでも /usr/local が作成できず、Homebrew 自体のインストールができない。

そのため、OSX 10.11 で Homebrew を使用する場合、インストールの前に  
/usr/local に関するパーミッションの設定や、必要に応じたシステム再起動を行う。

参考)

<http://qiita.com/riocampos/items/525ec4b35744ad586c5a>

# (補) OSX 10.11 での注意点

まず、/usr/local の有無を確認する。

Finder から [移動...] -> [フォルダへ移動...] と行い、移動先を /usr/local とする。  
Finder のwindowが開いたら、/usr/local は存在するが、「フォルダが見つかりません」というメッセージが表示されると、/usr/local は存在しない。

**/usr/local がある場合：**

Homebrew のインストールを試みる。

失敗する場合、パーミッションに問題があるので、下記のコマンドによりパーミッションを変更した後、Homebrewのインストールを試みる。

```
$ sudo chown $(whoami):admin /usr/local && sudo chown -R $(whoami):admin /usr/local
```

# (補) OSX 10.11 での注意点

/usr/local がない場合 :

- 1) リカバリモード (Command+R を押しながら起動) で再起動する。
- 2) ターミナルから、以下のコマンドを入力する。

```
$ csrutil disable
```

- 3) OSXを通常通りに再起動する。
- 4) ターミナルから、以下のコマンドを入力する。(まとめて1行)

```
$ sudo mkdir /usr/local && sudo chflags norestricted /usr/local &&  
sudo chown $(whoami):admin /usr/local &&  
sudo chown -R $(whoami):admin /usr/local
```

- 5) リカバリモード (Command+R を押しながら起動) で再起動する。
- 6) ターミナルから、以下のコマンドを入力する。

```
$ csrutil enable
```

- 7) OSXを通常通りに再起動し、Homebrewのインストールを試みる。