



# ROOT 講習会 第0回 (準備)

2021年 ROOT講習会 May 6 17:00--

Keita Mizukoshi (YMAP Kobe Univ.)

- ROOT講習会では以下を仮定します.
  - ターミナルソフトウェアが開けること
  - ROOTがインストールされていること
    - OSなどは問いません. 自分のローカルマシンにインストールしてあっても,ssh等で接続先にROOT環境があっても構いません.
  - 自分の好きなエディタが入っていること
- ROOT講習会では予習を推奨しますが以下を仮定しません.
  - 高度なshellコマンドの知識 (ターミナルを起動できればOK)
  - 高度なC++の知識
- 講習会(前/中/後)に(口頭/コメント/Slack)でいつでもわからないことを質問してください
  - 一度つまづくと追いつくのが大変なので

# コマンドライン

- 解析はコマンドラインで行うことが多いです。
- 端末,ターミナルなどと呼ばれるソフトウェアを起動します
  - WindowsだとWSL2 Ubuntu
  - LinuxだとGnome terminalなど
  - MacだとTerminal.appなど
- ここにコマンドを打ち込むことで作業します
  - 慣れると非常に便利です
  - マウスはあまりつかいません
  - 繰り返し作業や自動化を簡単に行うことができます



- プログラムを書く時に使うソフトウェア
  - Windowsのメモ帳などもエディタ
- コードを書く際に便利な機能がついているものを推奨
  - シンタックスハイライト
  - 補完機能
  -
- とりあえずAtomで始めるのがおすすめだが,エディタはなんでもいい
  
- **Sublime Text** 人気 一応有料
- **Atom** 人気 無料
- **VSCode** 人気
- CLion – エディタというより統合開発環境(IDE)
  - 初めはシンプルなエディタで始めるのがおすすめ
- Xcode – C++/rootの開発は不可能ではないが...
- Emacs – ショートカットを覚えて慣れればすごく便利
- Vi – Emacsと同じ 貧弱な環境でもほぼ必ず初めから入っている

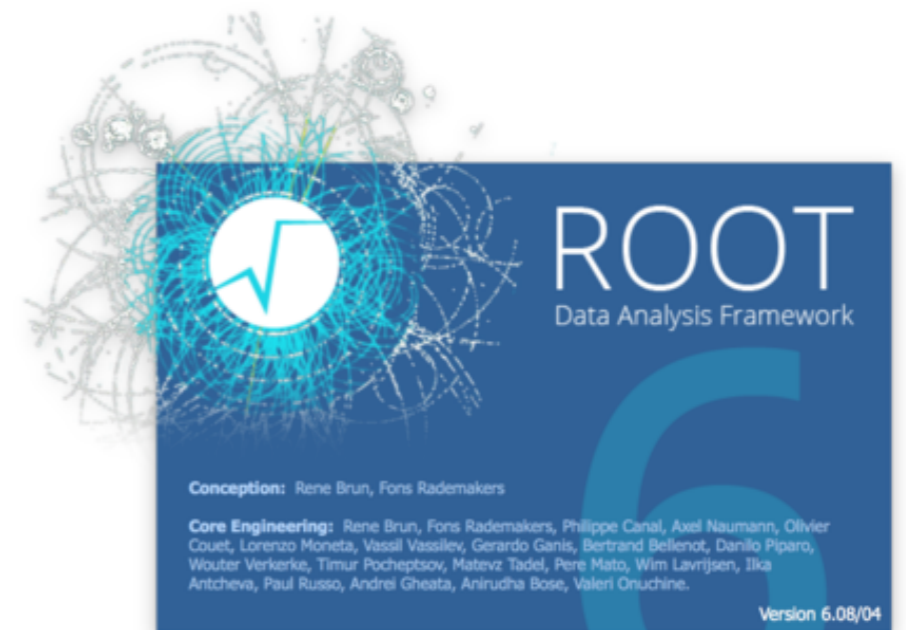
- Geant4ユーザーへのC++入門
  - <https://geant4.kek.jp/lecture/2016.11/materials/C++ForG4Users.pdf>
  - Geant4ユーザー以外にも有用です
- ATLAS C++ トレーニングコース
  - <http://www.icepp.s.u-tokyo.ac.jp/~sakamoto/education/atlasj/cplusplus/index.html>
  - ATLASの人以外にも有用です
- 講習会ではC++の専門的な知識を仮定しませんが,予習してあると理解が深まると思われれます.

- ROOT インストールの流れ
  - ROOTが依存しているソフトウェアをインストール
  - ROOTのソースコード をダウンロード
  - ROOTをコンパイル
  - ROOTにパスを通す
- Windows User 向け: [https://github.com/ymap-team/ROOT2021/wiki/Install\\_ROOT\\_Windows-Ubuntu-20.04](https://github.com/ymap-team/ROOT2021/wiki/Install_ROOT_Windows-Ubuntu-20.04)
- 講習会向けにサボる方法として,ROOTの入っているvirtual machineを使う方法があります.
  - Geant4 VM <https://wiki.kek.jp/display/geant4/Geant4+Virtual+Machine>
    - 放射線シミュレーションツールキットGeant4のvirtual machineですが,ROOTもはいています
- ROOTのインストールは初心者には難しい作業
  - あきらめずにがんばってみましょう

# ROOTインストールのゴール

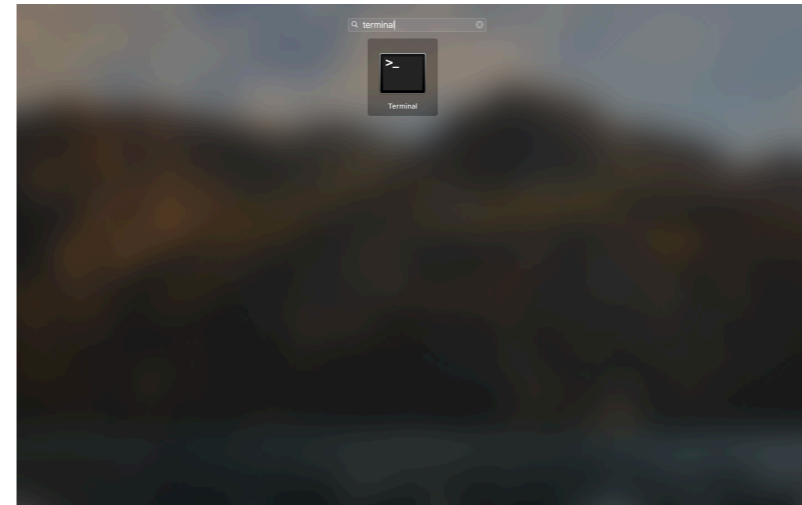
- 以下ができれば準備OK
- ターミナルで
  - `>root -b`とすると右のような表示が確認
  - `root [0]`と表示されると,
    - `.q`
    - と押すことで終了できる.
- `>root`とすると右のようなスプラッシュが表示される
- 注:先頭の`>`は自分で入力せずに初めからターミナルに表示されている
  - あなたの環境によって違う文字かもしれない
  - コマンドを打った後にエンターキーを押すことで実行される

```
> root -b
-----
Welcome to ROOT 6.16/00                                     https://root.cern
(c) 1995-2018, The ROOT Team
Built for macosx64 on Jan 23 2019, 09:06:13
From tags/v6-16-00@v6-16-00
Try '.help', '.demo', '.license', '.credits', '.quit'/.q'
-----
root [0]
```



# ROOT Install その1 (Mac)

- Terminal.appを起動します
- 'termianl'や'ターミナル'で検索
- 文字を打ち込める画面が出てきたら、まずいろいろなソフトウェアをいれるためのマネージャーhomebrewをインストールします
  - 必要なツールを一つ一つ調べて自力でインストールするのは面倒なので※
  - 下のコマンドをターミナルに貼り付けて実行してください
  - その後,説明に従ってインストールしてください.



※ソフトウェアAのインストールにBとCが必要になって,(依存関係という), BとCはそれぞれD,EとF,Gを必要として...という面倒を避けるためパッケージマネージャをつかえばAというだけで必要な全てをインストールしてくれる



```

mzks — zsh — 80x24
Last login: Fri Apr 17 18:22:11 on ttys006
[mzks@gold] ~
% /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
    
```

多少見た目は違っててもOK

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```



# ROOT Install その2 (Mac)

- 緑色で参加者が打ち込むコマンドを示す
- XCodeを念のためインストール
  - App Storeからインストール
  - その後,ターミナルでXCodeコマンドラインツールをインストール
- Homebrewで必要なものを入れる
  - cmake (ccmake) - コンパイル補助
  - XQuartz - 描画ツール(必須)
    - または,<https://www.xquartz.org> からダウンロードして導入してください
  - wget - ダウンローダー

ターミナルで作業

```
>sudo xcode-select --install
```

色々言われるので指示に従う

```
>brew install gcc
...(略)
```

```
>brew install cmake
```

```
>brew install wget
```

```
>brew cask install xquartz
```

※サボる方法 (余裕があれば非推奨)

```
>brew install root
```

これでインストールできてしまう

# ROOT Install その3 (Mac)

- ROOTをダウンロード
  - 公式サイト:<https://root.cern.ch>
  - 作業ディレクトリを作る
  - wgetで最新版をダウンロード
  - [https://root.cern/download/root\\_v6.22.08.source.tar.gz](https://root.cern/download/root_v6.22.08.source.tar.gz)

ターミナルで作業

```
>cd
>mkdir root_build
>cd root_build
>pwd
/あなたの環境のなにか/root_build

>wget https://root.cern/download/root_v6.22.08.source.tar.gz
``ダウンロード待ち...``

>tar xzvf root_v6.22.08.source.tar.gz
``展開される``
```

# ROOT Install その4 (Mac)

- tarで展開後,rootのソースコード がいっているディレクトリ (root-6.22.08など)がつくられているはず
- ROOTをコンパイルする用のディレクトリをつくる(build/)
- cdで'build'に入る
- cmakeでコンパイル用の設定をする
  - >cmake ../root-6.22.08 のあとにインストール場所を指定する
  - -DCMAKE\_INSTALL\_PREFIX=/opt/root/6.22.08 (=前後にスペースがないことに注意)
  - ここでは/opt/root/6.22.08にインストール
  - CMAKE\_CXX\_COMPILERには/usr/local/bin/g++-10
  - CMAKE\_C\_COMPILERには/usr/local/bin/gcc-10
    - brewで入れたコンパイラを指定する
    - mac 以外の方は >which g++ , >which gccで確認する
- make コマンドでコンパイル
  - PC性能に応じて待つ
- make install コマンドで完成物を配置する
  - /opt/など,一般ユーザーで書き込み権限がない場合は'sudo'をつける
- cmake && make && make installはROOTに限らず一般にUnixシステムにソフトウェアをインストールするイディオム

ターミナルで作業

```
>mkdir build
>ls
build/
root_v6.22.08.source.tar.gz
root-6.22.08

>cd build

>cmake ../root-6.22.08
-DCMAKE_INSTALL_PREFIX=/opt/root/6.22.08 -DCMAKE_CXX_COMPILER=/usr/local/bin/g++-10
-DCMAKE_C_COMPILER=/usr/local/bin/gcc-10
````
長い出力 自動設定中
うまくいけば最後に
-- Configuring done
-- Generating done
-- Build files have been written
to: /あなたの環境/build
と表示される
````
>make -j4
````結構待つ

>sudo mkdir -p /opt/root/6.22.08
>sudo make install
```

# ROOT Install その5 (Mac)

- インストールを確認
  - 先に指定したディレクトリにcdで移動
  - lsすると色々入っているのが見えるはず
- bin/thisroot.shがROOTの設定ファイル
- 'source thisroot.sh'で諸々設定される
- 'root -b'でrootが起動すれば成功

ターミナルで作業

```
>cd /opt/root/6.22.08
>ls
LICENSE      aclocal/    cmake/      emacs/      fonts/      icons/      lib/        man/         tutorials/
README/      bin/        config/     etc/        geom/       include/    macros/     test/
>cd bin
>ls
genreflex*      proofserv*      root.exe*      rootdrawtree*      rootn.exe*      roots.exe*      thisroot.csh
hadd*           proofserv.exe*  rootbrowse*    rooteventselector*  rootnb.exe*     rootslimtree*   thisroot.fish
hist2workspace*  rmkdepend*     rootcint*      rootls*             rootprint*      setenvwrap.csh*  thisroot.sh
memprobe*        root*           rootcling*     rootmkdir*          rootrm*          setxrd.csh       xpdtest*
prepareHistFactory*  root-config*   rootcp*        rootmv*             roots*           setxrd.sh

>source thisroot.sh

>root -b
```

# ROOT Install その6 (Mac)

- 毎回'source thisroot.sh'するのは面倒だ
- ターミナルを開いた時に自動で読み込まれるように設定する
- この設定ファイルは'.bashrc'という名前でホームディレクトリ(cdとだけ打った時に移動する場所)に作る
- あなたの好きなエディタで以下を.bashrcに追記する

```
1 # .bashrc - ROOT Setting
2 export ROOTSYS=/opt/root/6.22.08
3 cd $ROOTSYS/bin
4 source thisroot.sh
5 cd
```

注:

使っているshellの種類によっては, .bashrcではない  
コマンドラインで,

>echo \$SHELL

と打って/なんとか/bash となる場合は .bashrc に記述  
/なんとか/zsh となる場合は .zshrc に記述

または, エディタを使わずターミナルで作業する場合

```
> echo '# .bashrc - ROOT Setting' >> .bashrc
> echo 'export ROOTSYS=/opt/root/6.22.08' >> .bashrc
> echo 'cd $ROOTSYS/bin' >> .bashrc
> echo 'source thisroot.sh' >> .bashrc
> echo 'cd' >> .bashrc
```

- 追記予定
- ダメだったら1つ前のversionのROOTも試してみる

# エラー1

- エラーメッセージでぐぐると出てきたページ
- <https://root-forum.cern.ch/t/mac-os-x-gcc-6-compile-error/23727/2>
- CMake optionに以下を追加した.
- -Dcocoa=OFF -Dx11=ON  
-Dmysql=OFF

```
[ 48%] Built target Lzma
[ 48%] Built target Lz4
[ 48%] Built target Unix
[ 48%] Building CXX object
core/macosx/CMakeFiles/
Macosx.dir/src/
CocoaUtils.mm.o
cc1objplus: error: argument
to '-O' should be a non-
negative integer, 'g', 's' or
'fast'
make[2]: *** [core/macosx/
CMakeFiles/Macosx.dir/src/
CocoaUtils.mm.o] Error 1
make[1]: *** [core/macosx/
CMakeFiles/Macosx.dir/all]
Error 2
make: *** [all] Error 2
```