
Geant4インストール

Geant4ライブラリの作成

Geant4 10.3.P3 準拠

Geant4 HEP/Space/Medicine 講習会資料

本資料に関する注意

- 本資料の知的所有権は、高エネルギー加速器研究機構およびGeant4 collaborationが有します
- 以下のすべての条件を満たす場合に限り無料で利用することを許諾します
 - 学校、大学、公的研究機関等における教育および非軍事目的の研究開発のための利用であること
 - ・ Geant4の開発者はいかなる軍事関連目的へのGeant4の利用を拒否します
 - このページを含むすべてのページをオリジナルのまま利用すること
 - ・ 一部を抜き出して配布したり利用してはいけません
 - 誤字や間違いと疑われる点があれば報告する義務を負うこと
- 商業的な目的での利用、出版、電子ファイルの公開は許可なく行えません
- 本資料の最新版は以下からダウンロード可能です
 - <http://geant4.kek.jp/lecture/>
- 本資料に関する問い合わせ先は以下です
 - Email: lecture-feedback@geant4.kek.jp

本コースの目標

Geant4のソースパッケージとデータファイルをダウンロードしてインストールし、例題を実行して確かめる。

- 自分の目的に応じて環境を整備し、Geant4ライブラリをコンパイルしてインストールするために必要な条件を知り、その使い方を学ぶ。
- 物理データセットをインストールして、例題をコンパイルして実行し、可視化などの動作を確認する。
- アップデートやパッチへの対応

<http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/InstallationGuide/html/index.html>

「インストレーションガイドを読めばわかる」と言われても、、という方にも少しくどく説明します

目次

- 1.始める前に
- 2.プラットフォームの選択と前準備
- 3.Geant4ライブラリのコンパイルとインストールの準備
- 4.コンパイルとインストール実行
- 5.追加機能の利用
- 6.アプリが物理データセットを使うには
- 7.アプリがGeant4ライブラリを利用するには
- 8.アップデートへの対応
- 9.まとめと参照

〔謝辞〕 本講義で使用している資料は、過去にSLAC、CERN、IN2P3、ESAなどが主催したGeant4チュートリアルで使用されたスライドの内容を多く含みます。これらのスライド作成に寄与したGeant4 Collaborationメンバーに謝意を表します。

■ Geant4のインストール作業は避けて通れないこと

- ✓ 特定のプラットフォームで動くバイナリーコードを配布するシミュレーションコードと違い、Geant4は“Toolkit”であるシミュレーションソースコードを無償で配布している
- ✓ ユーザは自分の目的に合わせて配布物をコンパイルし、出来たGeant4ライブラリとファイル類一式を特定のディレクトリの下にインストールする作業が不可欠である。
- ✓ 自分が使うOSにGeant4を使うための外部ライブラリーや外部アプリを用意するのは必ずしも簡単な作業ではない。
 - ✓ とりわけ、それらもバージョンアップされるので、Geant4のReleaseNoteを注意深く読み、自分のパソコンの対応について調べねばならない。

■ Geant4アプリケーションは自分で作るもの

- ✓ Geant4ライブラリにリンクして実行する自分のアプリケーションC++コードを作成せねばならない
- 自分のアプリがGeant4ライブラリを使うためにはcmakeが必要とするCMakeLists.txtを準備する
- 先ずはGeant4の例題を学ぶのが近道
- 本講習会のCMakeLists.txtは上のものを使いやすく改善している

Geant4を維持管理する人へ

■ インストールへの関わり方

- ✓ 自分のパソコンへインストールせねばならない場合、本コースの内容を理解して実習すれば大丈夫。
- ✓ チームの誰かがすでにインストールしてくれているものを使うだけの場合であっても、
 - ✓ ライブラリに組み込まれている可視化など多数の外部パッケージが自分の利用に合っているかどうかを判断したり、
 - ✓ 並列処理ができるかどうか、
 - ✓ 物理データセットの格納状況、など、既存のインストール状況を知っておく必要がある。不足の場合や古いバージョンやパッチが当たっていない場合などは、あなたがインストールしてみんなに使って貰えば良い。

■ 後で述べるが、本格的な使用を始めたら、その期間にはGeant4のアップデートとユーザアプリのアップデートが起こりうる。

- ✓ 現在のGeant4ライブラリのインストール状況の確認をして、
- ✓ 複数のバージョンのインストレーションがあるなら、アプリがそれらを選んで効率的に使い分けるにはどうすればよいか

プラットフォームの選択と前準備

プラットフォーム = {OSとコンパイラ}

- 公式にサポートされているプラットフォーム
 - プラットフォームとは、OSとコンパイラの組み合わせのこと
 - ✓ サポートされているプラットフォームは、ユーザドキュメント”Installation Guide”の第一章を調べると良い。
 - ✓ 最新のより詳しい情報は”10.3 ReleaseNote”参照のこと
- Unix系
 - 64bit Linux: Ubuntu, Scientific Linux, などと gcc-4.8.5 以上: **Linux-g++**
 - ✓ C++-11 標準のサポート条件についてはガイドの1.2を参照
 - macOS
 - ✓ OSX(10.12 Sierra) と clang3.8 以上 : **Darwin-g++**
 - ✓ 統合開発環境 Xcode を使うには、command line tools を追加インストールすること。
- Windows 10 (multi-threadには対応せず)
 - 統合開発環境 Visual C++ 14.0 (Visual Studio 2015): **WIN32-VC**
- 本講では今後、Linux をプラットフォームとして Geant4 のライブラリ作成を説明する。macOS + Xcode もほぼ同じ。

ビルドツール cmake

- CMake (= cross-platform makefile generator) 以前
 - ✓ Geant4.9までは、様々な外部パッケージを選択してコンパイルするための環境変数を予め設定しておく必要があった。
 - ✓ さらに、ユーザアプリケーションのコンパイル時には、アプリが使用する外部パッケージを選ぶ環境変数を設定する必要があった。
 - ✓ これらの環境変数はプラットフォーム毎に違った名前を与えねばならなかったので、ユーザの利用環境を設定するシェルスクリプトは複雑なものになった
- CmakeはLinux, OSX, Windowsなどのプラットフォームに依存しないビルドツール
 - ✓ Geant4.10でcmakeが採用され、ライブラリ作成に必要なオプションをディレクティブとして与えてcmakeを実行すれば、ライブラリを作成するためのコンパイル手順(makefile)が、プラットフォームを指定せずに自動的に作られるようになった。
 - ✓ ライブラリのコンパイル時には環境変数を指定する必要がなくなった。
 - ✓ 可視化、Multi-thread, GDML, などもプラットフォームによらずコンパイルできる
- cmakeに与えるディレクティブは多数あるが
 - ディレクティブ全部の説明は”Installation Manual”を読むこと
 - ライブラリのインストール先(必須)
 - マルチスレッドの選択(選択)
 - Vis / GUI オプション(選択)
 - 必須のもの他に選択ものでも既定値が使えるかどうかなど、マニュアルに

cmakeの点検／更新／導入

■ Linux

- ディストリビューションに既に含まれていることが多いので、既に入っているかどうか、バージョンは大丈夫かどうかをチェック
 - ✓ `Linux: ~ % cmake --version`
 - ✓ `cmake version 3.5.2` > 3.3 = Geant4.10.3の要求バージョン
 - Linuxのパッケージ管理ツールを使ってバージョンアップする
 - ✓ `%sudo apt-get install cmake` (Debianの場合)
 - ディストリビューションのバージョンがこの要求よりも古いなら
 - ✓ CmakeウェブサイトからLinux i386版をインストールするとよい
 - MacOSX
 - ✓ XcodeをAppStoreからインストールする
- ## ■ おまけ cmakeの代わりに使えるccmake 使用例は後述
- Linux : `sudo apt-get install cmake-curses-gui`
 - OSX : Cmake と一緒にインストールされる
 - ✓ `%which ccmake; ccmake -version`

Geant4ライブラリの コンパイルとインストールの準備

最新のGeant4 ツールキットと物理データセット

- Geant4ソースコード 現在のものは10.03.p03
 - メジャーリリース：現在は10
 - ✓ 新しい機能が追加されたり、性能が向上するだけでなく、
 - ✓ 例えば継承の仕方やpublic関数のシグネチャが変化または廃止など、以前のメジャーバージョンとの互換性が無くなることもありえる
 - ✓ 添付されるリリースノートを必ず読むこと
 - マイナーリリース：現在は3
 - ✓ (原則として!) 前のマイナーリリースとの互換性を保ちつつ、新しい機能の追加や陳腐化した機能への警告メッセージなどの改良がされる。
 - ✓ ほぼ毎年マイナー番号が上がり、本年末には10.4が予定されている
 - パッチバージョン：現在はpatch-03
 - ・ リリース公開後に、パッチを含んだ全ソースコードを公開することがある
 - 配布されているVMには最新の10.3.patch03がインストールされている。
- 物理データファイルセット
 - ライブラリー作成時には無くても良い。アプリの実行時に必要となる。
 - ✓ だが、ライブラリー作成時にファイルをダウンロードすると、アプリの環境設定が楽になる

作業用と成果物を格納するディレクトリを決める

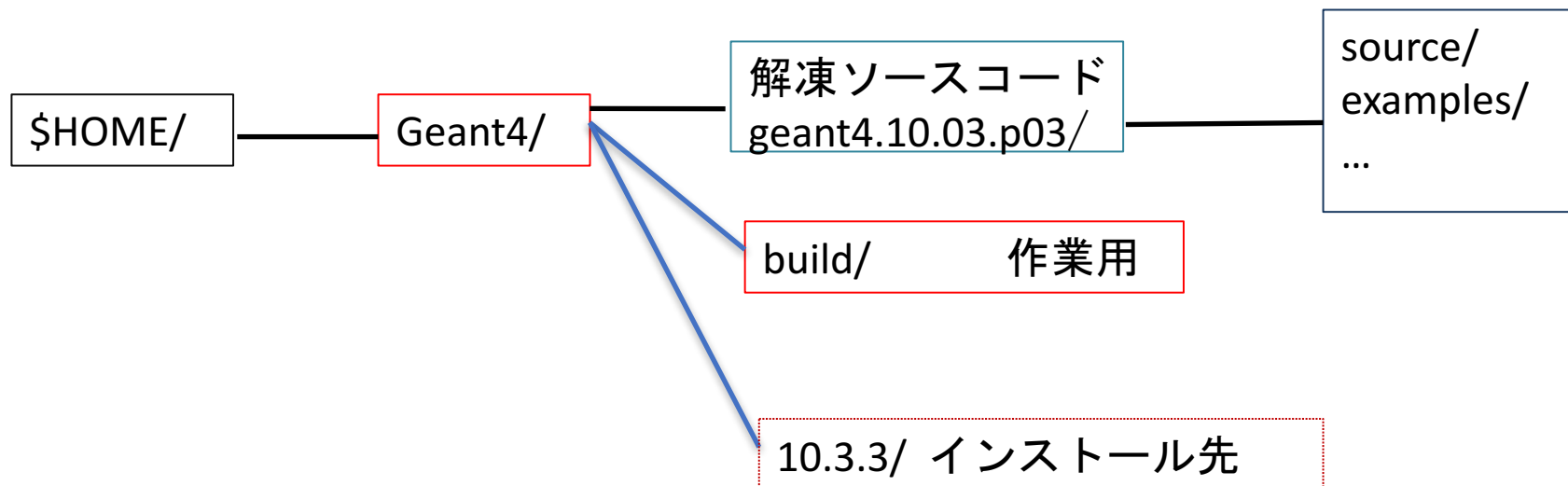
- Geant4ライブラリを作成し保管するために3つのディレクトリを用意する
 1. Geant4のソースコードを解凍するソースコード格納ディレクトリ
 2. コンパイル作業用のビルドディレクトリ
 3. 作成したライブラリとヘッダファイルを格納するインストールディレクトリ.
- 自分だけの利用ならホームディレクトリの下に置けばパーミッションの操作不要。
以下の説明では~/Geant4/を使うものとする。
 - ソース ~/Geant4/geant4.10.03.p03/: ダウンロードしたtarball geant4.10.03.p03.tarを展開したら自動的に作られる。この内容は変更されない
 - ビルド ~/Geant4/build/: ビルド作業を行うところで、最初は空っぽ。インストールが終わったら不要だが、再利用のために保存するのが良い。
 - インストール ~/Geant4/10.3.3/: ビルド成果を置く場所。次の3つのディレクトリが自動的に作成される
 - ✓ ~/Geant4/10.3.3/lib, include
 - ✓ ~/Geant4/10.3.3/share
 - ✓ ~/Geant4/10.3.3/bin

Tips: チームのためにインストールする場合: チームメンバーがアクセス出来る場所としてたとえば/opt/geant4 の下にGeant4インストールディレクトリを置き、グループメンバーに読む権利を与える。本講習会で配布している仮想マシンでは以下のようである。

✓ /opt/geant4/10.3.3

Cmakeの前準備作業の例

- ダウンロードした圧縮Geant4ソースファイルを\$HOME/Geant4の下に置く
 1. `%cd; mkdir Geant4`
 2. `%mv ~/Downloads/geant4.10.03.p03.tar.gz ~/Geant4`
 3. `%cd Geant4; tar -fvxz geant4.10.03.p03.tar.gz`
 - `~/Geant4/geant4.10.03.p03/`の下にソースが展開される
- `%mkdir build` 作業用ディレクトリ
- `%mkdir 10.3.3` インストールディレクトリ



Cmakeのビルドディレクティブを決める

- Geant4に含めたい機能や、ライブラリーの構成方法などについては、cmakeのコマンド行でdirective (形式は `-DXXXXXX=YYYYYY`)直接指定する。以下のディレクティブは、最小限使えるようにしたい機能だけに限定している。
 - ✓ `-DCMAKE_INSTALL_PREFIX=`
✓ (既定値: `/usr/local`) 自分だけなので `$HOME/Geant4/10.3.3/` とする
 - ✓ `-DGEANT4_INSTALL_DATA = ON (DEFAULT : OFF)` makeの時に自動的にダウンロードする
 - ✓ `-DGEANT4_INSTALL_DATADIR` 上のがONなら指定不要で下の既定が使われる
✓ (既定値: `CMAKE_INSTALL_DATAROOTDIR = share/`)
 - ✓ `-DGEANT4_USE_OPENGL_X11=ON` (既定値 : OFF, Unix のみONにできる)
 - ✓ `-DGEANT4_USE_SYSTEM_EXPAT` (既定値 : ON) 無ければ入れること

可視化外部パッケージ OpenGLを使う準備

■ OpenGL開発パッケージ

- ◆ Linuxでは見るためにOpenGLがインストールされているが、OpenGLを使うアプリをコンパイルするには更にヘッダファイルなど開発パッケージが必要となる。
 - ◆ Cmakeでエラーメッセージが出る場合には、次のようなものをインストールする必要がある場合がある。
 - ✓ libx11-dev, libxt-dev, libgl-mesa-dev, libxmu-dev, libglx-dev, libglut-dev (名前は少し違っているかもしれない)
 - ✓ ヘッダファイル (通常パッケージ名の後ろに“-dev”または“-devel”) が無ければエラーメッセージが出る。
 - ✧ Linuxではパッケージ管理ツールを使って足りないものを追加インストールすることを勧める。
 - ✧ Linuxではグラフィックスカードのアクセラレータドライバもインストールする。これも通常はパッケージ管理ツールでできる。
- macOS ではXcode

データセットの自動インストール設定を選んで見る

- Cmakeの機能を使い、Geant4ではすべてのデータファイルを一括ダウンロードして所定の場所に解凍するためのオプションを提供している
 - ✓ -DGEANT4_INSTALL_DATA = ON (default OFF)
 - ✓ 既定の格納場所は CMAKE_INSTALL_PREFIXの下のshareの中になる
\$HOME/Geant4/10.3.3/share/Geant4-10.3.3/data
 - ✓ データセットはライブラリ作成には不要で、アプリを実行するときに要求される。
上のディレクティブを指定すると、データセットの環境変数を設定するためのスクリプトが自動的に\$HOME/10.3.3/bin/geant4.csh(sh)という名前で作られる
- ◆ 自動ダウンロードで起きるかもしれない不具合
 - ✓ コンピュータ環境（ネットワークやディスククォータなど）によってはデータの取得に失敗するケースがHyperNewsで報告されている。例えば次のような場合、やり直せば成功することもある。

error: downloading 'http://geant4.cern.ch/support/source/G4NDL.4.0.tar.gz' failed
status_code: 28
status_string: "Timeout was reached" 時間切れの既定値は1500秒

コンパイルとインストール実行

cmake : ビルド実行

```
% cmake -DCMAKE_INSTALL_PREFIX=/Users/yoshida/Geant4/10.3.3¥ ← 絶対パス
-DGEANT4_USE_OPENGL_X11=ON¥
-DGEANT4_INSTALL_DATA=ON¥
../geant4.10.03.p03 ← ソース指定は相対パスでよい

-- Configuring download of missing dataset G4NDL (4.5)
-- Configuring download of missing dataset G4EMLOW (6.50)
-- Configuring download of missing dataset PhotonEvaporation (3.2)
-- Configuring download of missing dataset RadioactiveDecay (5.1.1)
-- Configuring download of missing dataset G4NEUTRONXS (1.4)
-- Configuring download of missing dataset G4PII (1.3)
-- Configuring download of missing dataset RealSurface (1.0)
-- Configuring download of missing dataset G4SAIDDATA (1.1)
-- Configuring download of missing dataset G4ABLA (3.0)
-- Configuring download of missing dataset G4ENSDFSTATE (2.1)
-- The following Geant4 features are enabled:
GEANT4_BUILD_CXXSTD: Compiling against C++ Standard '11'
GEANT4_USE_SYSTEM_EXPAT: Using system EXPAT library
GEANT4_USE_OPENGL_X11: Build Geant4 OpenGL driver with X11 support
中略
-- Configuring done
-- Generating done
-- Build files have been written to: ~/Geant4/build
```

データセットはコンパイル時に
取得してインストール先の
10.3.3/share/Geant4-10.3.3/data/
に展開される

Build/には沢山のファイルとディレクトリが
自動的に作られている。

make: コンパイルとインストール実行

```
[MacBookPro:~/Geant4/build] yoshida% make -j4 コンパイル開始
[ 0%] Built target RealSurface
[ 0%] Built target G4NEUTRONXS
[ 0%] Built target G4ENSDFSTATE
[ 0%] Built target G4PII
[ 0%] Built target RadioactiveDecay
[ 0%] Performing download step (download, verify and extract) for 'G4EMLOW'
...中略
-- downloading...
   src='http://geant4.cern.ch/support/source/G4EMLOW.6.50.tar.gz'
   dst='/Users/hajimeyoshida/Geant4/build/Externals/G4EMLOW-6.50/src/G4EMLOW.6.50.tar.gz'
   timeout='1500 seconds'
[ 4%] Built target G4gl2ps
[ 6%] Built target G4global
[ 6%] Built target G4intercoms
[ 8%] Built target G4materials
..中略
[100%] Linking CXX shared library ../../BuildProducts/lib/libG4physicslists.dylib
[100%] Built target G4physicslists      すべてのモジュールのコンパイルが無事終了
[MacBookPro:~/Geant4/build] yoshida% make -j4 install   インストールして終了
```

成果： インストールディレクトリの構成

macOS 10.13 の例

10.3.3	
名前	種類
▼ bin	フォルダ
geant4.sh	Shell Script
geant4.csh	TextWrang...t document
geant4-config	Unix executable
▼ include	フォルダ
▶ Geant4	フォルダ
▶ lib	フォルダ
▼ share	フォルダ
▼ Geant4-10.3.3	フォルダ
▶ data	フォルダ
▶ examples	フォルダ
▶ geant4make	フォルダ
Geant4-10.3.3	書類
tools.license	書類

データ設定環境変数
設定スクリプト

Geant4クラス定義ファイル

Geant4モジュールライブラリ

データファイル

ライブラリ作成についてより細かくチューニングするには

ライブラリのインストールに関するオプション

- ✓ BUILD_SHARED_LIBS : (DEFAULT : ON)
- ✓ BUILD_STATIC_LIBS : (DEFAULT : OFF)
- ✓ CMAKE_INSTALL_BINDIR : (DEFAULT : bin)
- ✓ CMAKE_INSTALL_INCLUDEDIR : (DEFAULT : include)
- ✓ CMAKE_INSTALL_LIBDIR : (DEFAULT : lib(+?SUFFIX))
- ✓ CMAKE_INSTALL_DATAROOTDIR : (DEFAULT : share)
- ✓ CMAKE_BUILD_TYPE (DEFAULT : Release)
- ✓ GEANT4_INSTALL_DATA_TIMEOUT : (DEFAULT : 1500)
- ✓ etc.

普通はどれも既定値のままでよい。
変えたい場合はマニュアルを読んで
確認すること

追加機能の利用

追加機能を組み込んだライブラリを作る

- 以下の機能を使えるGeant4ライブラリをコンパイルするには外部ライブラリを必要とする。これらが正しくインストールされていたら、次のスライドのようにオプションディレクティブを指定するとよい。
 - データ永続性(persistence)
 - ✓ GDML (Geant4 geometry Description Markup Language) : Xerces-C++ header files & libraries
 - 可視化(visualization)
 - ✓ GL: OpenGL/MesaGL
 - ✓ Qt: Qt4/5 header files & libraries + OpenGL/MesaGL
 - ✓ DAWN: 不要
 - ✓ RayTracer : OpenGL/MesaGL
 - データ解析(analysis)
 - ✓ ROOT, CSV, : 不要
- 並列処理(parallel processing)
 - ✓ Linuxのみ、コンパイラはThread Local Storage, pthread libraries and headers
 - ✓ 配布のVMにはコンパイル済みのライブラリー/opt/geant4/10.3.3-mt/ を置いている。

追加機能を組み込むためのビルドオプション

- GEANT4_BUILD_MULTITHREADED : (DEFAULT : OFF, Unix Only)
- GEANT4_USE_GDML : (DEFAULT : OFF | ON にするにはXERCESC_ROOT_DIRが設定されていること)
- GEANT4_USE_QT (DEFAULT : OFF)
- GEANT4_USE_OPENGL_X11 (DEFAULT : OFF, Unix Only)
- GEANT4_USE_RAYTRACER_X11 (DEFAULT : OFF, Unix only)
- GEANT4_USE_SYSTEM_EXPAT (DEFAULT : ON)
- GEANT4_USE_SYSTEM_ZLIB (DEFAULT : OFF)
- etc.

既定値を変える場合には、必要な開発用外部パッケージを確認すること

アプリが物理データセットを使うには

物理データセット

■ 用途

- ✓ 物理データセットはアプリを実行するときに必要となる
- ✓ 環境変数でそれへのパスを通していない時は実行時にエラーとなる
- ✓ `.bashrc`に設定スクリプトを書いておくとも便利。ただし、自動的に生成されたスクリプトはディレクトリの位置を決められないので、次のように指定してあげる。

```
cd ~/Geant4/10.3.3/bin ; source geant4.sh
```

■ データセットの手動インストールをしたい場合

- Geant4ライブラリのコンパイルが終わった後でも手作業でインストール出来る
 - 既にインストールしてあるデータセットの一部だけの更新や追加をしたいとき必要なデータファイルをCERNのWebページからダウンロード

■ 既にデータを置いてあるなら自分で格納場所を指定するだけでcmake

- ✓ パッチが出てもデータはそのままでも使えることがあるので、自分の場所を使って、
 - ✓ `-DGEANT4_INSTALL_DATA = OFF`
 - ✓ `-DGEANT4_INSTALL_DATADIR = $HOME/Geant4/G4DATA`

■ 環境変数設定の自前のスクリプトを用意する。

- ✓ Bash/cshで書き方が少し違うけれど、

```
setenv G4RADIOACTIVEDATA "~/Geant4/G4DATA/RadioactiveDecay5.1.1"
echo "On this machine the G4RADIOACTIVEDATA=$G4RADIOACTIVEDATA"
```

データセットへのパスを確認する

- GEANT4_INSTALL_DATA=ONなら下のgeant4.csh(geant4.sh)が自動的に作られるのでそれをsourceするだけでよい。

```
yoshida% source ~/Geant4/10.3.3/bin/geant4.csh
```

```
yoshida% env | grep G4      確認
```

```
G4NEUTRONHPDATA=/Users/yoshida/Geant4/10.3.3/share/Geant4-10.3.3/data/G4NDL4.5
```

```
G4LEDATA=/Users/yoshida/Geant4/10.3.3/share/Geant4-10.3.3/data/G4EMLOW6.51
```

```
G4LEVELGAMMADATA=/Users/yoshida/Geant4/10.3.3/share/Geant4-10.3.3/data/PhotonEvaporation3.2
```

```
G4RADIOACTIVEDATA=/Users/yoshida/Geant4/10.3.3/share/Geant4-10.3.3/data/RadioactiveDecay5.1.1
```

```
G4NEUTRONXSDATA=/Users/yoshida/Geant4/10.3.3/share/Geant4-10.3.3/data/G4NEUTRONXS1.4
```

```
G4PIIDATA=/Users/yoshida/Geant4/10.3.3/share/Geant4-10.3.3/data/G4PII1.3
```

```
G4REALSURFACEDATA=/Users/yoshida/Geant4/10.3.3/share/Geant4-10.3.3/data/RealSurface1.0
```

```
G4SAIDXSDATA=/Users/yoshida/Geant4/10.3.3/share/Geant4-10.3.3/data/G4SAIDDATA1.1
```

```
G4ABLA3DATA=/Users/yoshida/Geant4/10.3.3/share/Geant4-10.3.3/data/G4ABLA3.0
```

```
G4ENSDFSTATEDATA=/Users/yoshida/Geant4/10.3.3/share/Geant4-10.3.3/data/G4ENSDFSTATE2.1
```

パスが通っていない時は実行時にコアダンプする。

```
----- EEEE ----- G4Exception-START ----- EEEE -----
```

```
*** G4Exception : PART70000
```

```
    issued by : G4NuclideTable
```

```
G4ENSDFSTATEDATA environment variable must be set
```

```
*** Fatal Exception *** core dump ***
```

```
----- EEEE ----- G4Exception-END ----- EEEE -----
```

Cmakeの代わりになるツール : ccmake

ライブラリをbuildしたディレクトリの情報を使ってccmakeを実行するとオプションが何であったかがわかる

```
%ccmake $HOME/Geant4/build
```

```
ターミナル — ccmake . — 80x24

Page 1 of 2

CMAKE_BUILD_TYPE
CMAKE_INSTALL_PREFIX
CMAKE_OSX_ARCHITECTURES
CMAKE_OSX_DEPLOYMENT_TARGET
CMAKE_OSX_SYSROOT
GEANT4_BUILD_MULTITHREADED
GEANT4_INSTALL_DATA
GEANT4_INSTALL_DATADIR
GEANT4_USE_G3TOG4
GEANT4_USE_GDML
GEANT4_USE_INVENTOR
GEANT4_USE_OPENGL_X11
GEANT4_USE_QT
GEANT4_USE_RAYTRACER_X11
GEANT4_USE_SYSTEM_CLHEP
GEANT4_USE_SYSTEM_EXPAT
GEANT4_USE_SYSTEM_ZLIB

Release
/Users/hajimeyoshida/Geant4-10/10.3.3
/Applications/Xcode.app/Contents/Developer/Pl
OFF
OFF
OFF
OFF
OFF
ON
OFF
OFF
OFF
OFF
ON
OFF

GEANT4_USE_SYSTEM_ZLIB: Use system zlib library
Press [enter] to edit option
Press [c] to configure
Press [h] for help
Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)

CMake Version 3.5.2
```

データファイルは既存
のものを使い回す

OpenGLを使う

Geant4ライブラリの利用

講義「cmakeと演習プログラムのビルド」の復習を少しだけ

アプリのCMakeLists.txt

- cmakeを使うアプリケーションの書き方: Installation Manual 3.2参照
 - アプリケーションはcmakeのfind_packageコマンドを使ってGeant4のインストールディレクトリにあるコンフィグレーションファイルGeant4Config.cmakeを読み込む
 - そのためのスクリプトファイルCMakeLists.txtをアプリケーションのディレクトリの下に作っておかねばならない
 - コンパイル用に build ディレクトリを用意してそこでcmakeする
 - Geant4Config.cmakeを格納しているディレクトリを、アプリのビルド時に明示的にオプション指定する
 - ✓ `%cmake -DGeant4_DIR = /opt/geant4/10.3.3/lib/Geant4-10.3.3 App_source_Dir`

注意 Geant4の例題で提供されているCMakeLists.txtは実際の使用ではマクロファイルがbuild/にコピーされるなど不便な点がある。

本講のハンズオンではアプリの実行コードの置き場所はbin/とし、マクロを使って実行するためにはTestBench/を使うなどの改良したものを使っている。比較されたし。

ライブラリのGeant4Config.cmake

- インストールディレクトリ~/Geant4/10.3.3/lib/Geant4-10.3.3の下にコンフィグファイルGeant4Config.cmakeが置かれている。ディレクトリ名にバージョン番号が自動的に追加されていることに注意。

[MacBookPro:~] yoshida% ls ~/Geant4/10.3.3/lib/Geant4-10.3.3/

Darwin-clang

Geant4Config.cmake

Geant4ConfigVersion.cmake

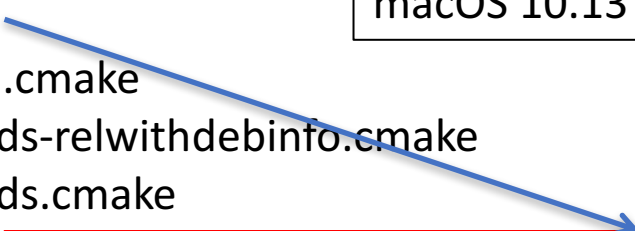
Geant4LibraryDepends-relwithdebinfo.cmake

Geant4LibraryDepends.cmake

Modules

UseGeant4.cmake

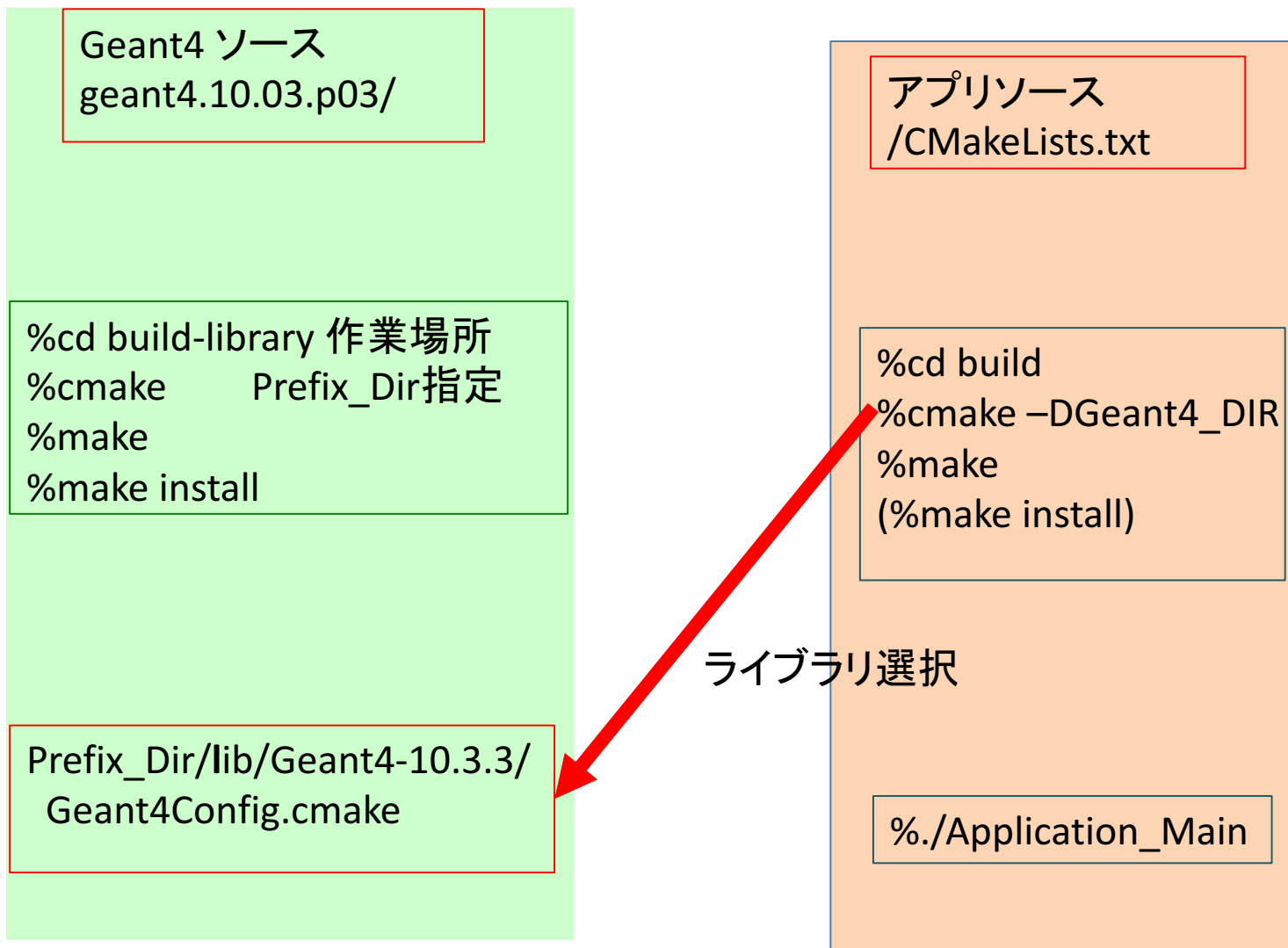
macOS 10.13 + Xcode 9.0.1+ Xquartz 2.7.11 の場合



```
# - Geant4 CMake Configuration File for External Projects
# This file is configured by Geant4 for use by an external project
# This file is configured by Geant4's CMake system and SHOULD NOT BE
EDITED
#
# It defines the following variables
# Geant4_INCLUDE_DIRS - include directories for Geant4
# Geant4_DEFINITIONS - compile definitions needed to use Geant4
# Geant4_LIBRARIES - libraries to link against
# Geant4_CXX_FLAGS - Recommended CXX flags for the compiler used to
# build the Geant4 libraries. 以下長文
```


アプリがライブラリを選ぶには

Geant4ライブラリーとそれを使うアプリの関係図



アプリで追加機能を使うには

■ アプリが使うために必要な方法は二通り

1. Geant4ライブラリに組み込み済みならば、アプリ側では指定なしに使える。
 - GDML, OpenGL, Qt, RayTracerなど
2. Geant4ライブラリ作成時にオプションディレクティブが不要なDAWN, Analysisでは
 - ✓ 外部アプリ(DAWN, ROOT など)のインストールが別途必要
 - ✓ DAWNでは可視化のマクロファイルを用意し、独立したDAWNアプリで描画
 - ✓ 解析では外部ツールが何であれ、アプリが定義ファイルに使用を指示する

アップデートへの対応

ライブラリとデータセットのアップデート

- Geant4 ツールキットのアップデートでは
 - マイナーアップデートやパッチ（今後10.3 -> 10.4）
 - ✓ APIは変更されないので自分のアプリはそのまま使い続けられる
 - ✓ Release Noteを読んで必要ならばGeant4ライブラリを作りなおす
 - ✓ 自分のアプリの再コンパイルは必要
 - メジャーアップデート（過去9.6 -> 10.0）
 - ✓ APIは後方互換性がなく、自分のアプリは新しいGeant4では動かず、ソースコードを変更せねばならないかもしれない
 - ✓ Release Noteをよく読んで決めるしかない
- Geant4のバージョンアップでどうするか
 - バージョン別にビルドとインストールディレクトリを作る。
たとえば、/opt/geant4/10.2.2 => /opt/geant4/10.3.3
 - ✓ アプリが最新のバージョンにうまく移行できるかどうかを確認できるまでは、現行のバージョンを使っているアプリのバイナリーも保存するとよい
 - アプリの異なるバージョンがあっても混乱せずに管理できる
 - ✓ アプリのバージョン管理も必要になるだろう
- データセットの更新
 - ✓ 環境変数の再設定とアプリの再コンパイルだけでよい

複数のライブラリを使い分ける

- ✓ 共有(shared)ライブラリが標準なので、アプリの実行時には使うものだけがリンクされる。
- ✓ だが、オプションが違う複数のライブラリを使い分けたい時には、各々にビルドとインストールディレクトリの対を用意する。
- ✓ 例えば、CMAKE_BUILD_TYPE を最適化を行う既定値の”Release”のビルドと、デバッグ記号を付ける”Debug”のビルドを作る。
- ✓ 一度cmakeしておけば、ソースコードを変更してもビルドしなくて良い。Makeするだけ。

まとめ

Geant4 Installation Guideを読もう

“Building and Installing Geant4 for Users and Developers”

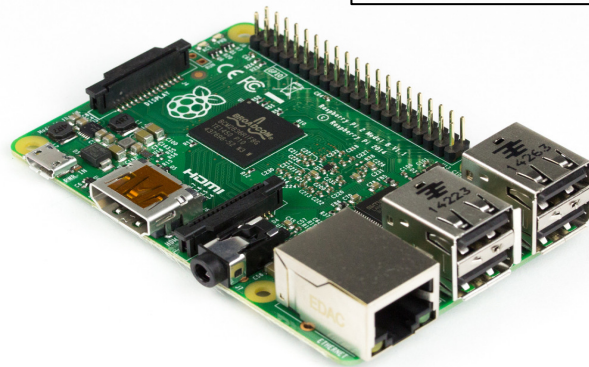
1. Geant4のビルドとインストール
 2. Geant4を使うには
 3. Geant4アプリ開発者のためのCmakeとビルドツール
-
- ✓ 追加機能を入れるためのオプション指定の方法はこれに網羅されている
 - ✓ アプリをコンパイルするためのCmakeList.txtを書くための詳しい説明がある
 - ✓ Windowsを使いたい場合はこれを読むしかない。
 - ✓ 新しいマイナーリリースがあると、一部古い記述がある。そのときはReleaseNoteをよく読むと良い
 - ✓ 同じような問題を持っているユーザがいないかどうか、HyperNewsを探してみる
<http://hypernews.slac.stanford.edu/HyperNews/geant4/get/installconfig.html>

新しいプラットフォームに挑戦する人は？

- Windows Subsystem for Linux
 - Windows 10 Fall Creators Update, 17 Oct. 2017
 - OSS 開発環境としてmacOSに対抗したい？
 - Ubuntu, OpenSUSE, Fedora
 - Xserver, X11 clients
- KEKのS氏によると
 - Geant4のコンパイルはできた。
 - Qtが不安定なようだ
- 日本マイクロソフトのWSLエヴァンジェリスト
 - 「試してみて、不具合があればGitHubへ報告して欲しい、追加して欲しい機能もリクエストして欲しい」と(日経Linux 2017 11月号)
- Windows10を持っている研究室は多いのでは。
 - VMよりもコンピュータ資源への要求が少ないかも

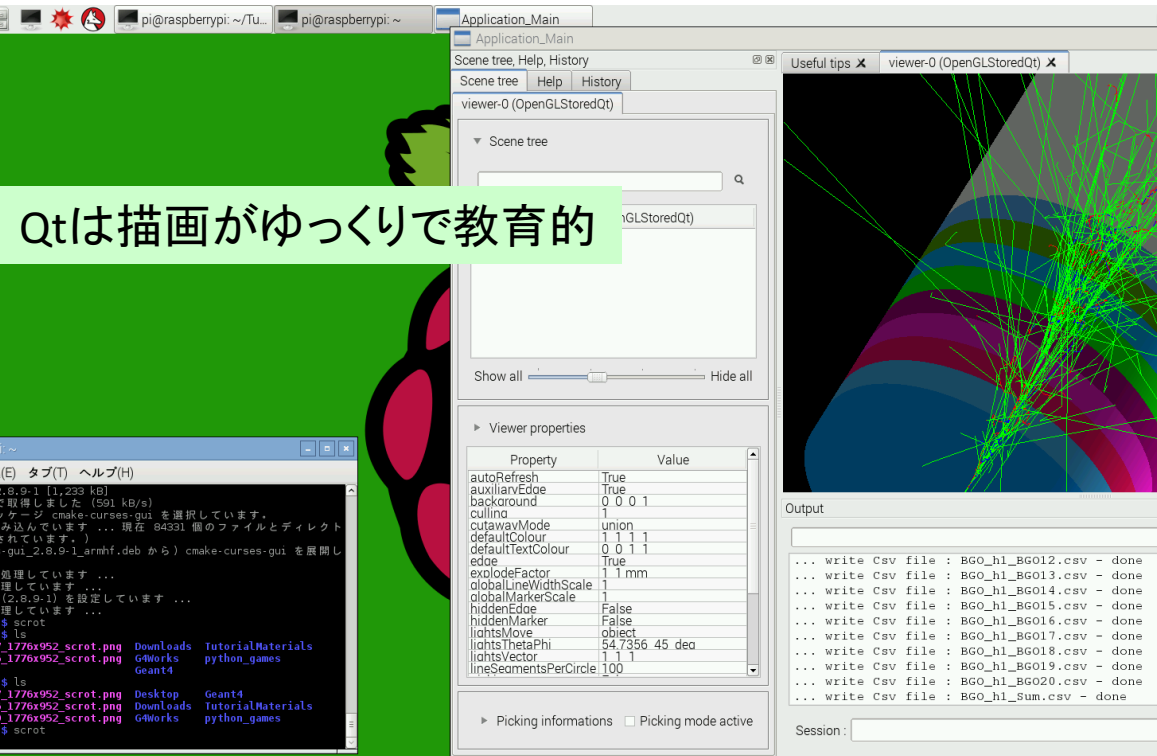
おまけ ラズパイ遊び： Geant4が走る最小最安値のシステム

本稿の方法でゼロインストール



Raspberry Pi 2 B £30

g++, make, cmake, ccmake, Qt4, GL



A 900MHz quad-core ARM Cortex-A7 CPU
1GB RAM
Full Linux a la Debian
コンパイルに一晩かったが、Geant4
のフル機能が使える

BGO_h1_Sum.csv
Raspberry Pi 2B

