

経過報告

田中 隆己

2010年10月19日

1 報告

1.1 新たな決まり

今回のミーティングより、各自が作成した資料をまとめたファイルを作ることになった。そして、回収担当は4年生となった。つきましては、今後余分に一枚印刷お願いします。

1.2 アップロード

- ・ DEMONS を使ったログのまとめを前回作成したが、それは自分の web にアップロードした。
- ・ 研究会のまとめはひとまず送ってもらった内容をそのまま web に貼り付けた。問題があれば言ってください。会場、特に無線 LAN の申請や教室の申請、鍵の管理などのまとめを書いてくださる方募集中です。

2 smallNEUT(仮)

2.1 見積り

smallNEUT の前実験の見積りが揃ったので以下に示す。

・ PMT(浜フォト)

H7415 77,400 円 * 4

R4124ASSY 63,000 円 * 2

計 457,380 円 (税込)

・ シンチレータ関連 (ジーテック)

EJ-200 20*40*1000 36,000 円 * 2

EJ-200 20*20*1000 28,000 円 * 2

EJ-200 10*10*1000 20,000 円 * 1

EJ-200 5* 5*1000 22,000 円 * 1

optical grease(チューブ入り) 30,000 円 * 1

アルミナイズドマイラー 1000*1200[mm] 0 円 (頂ける) * 1

計 210,000 円 (税込)

総計 667,380 円

以上の内容で既に発注しており、シンチレータ関連は今日から 2 週間程度、PMT が 11 月 24 日に発送予定です。

2.2 実験内容

主に、宇宙線を使って時間分解能のエネルギー（発光量）依存性と発光量の形状依存性を調べる。実は、これに近い実験をしている論文があるのでそれと比較しながら実験を進める。（論文では一部レーザーを使って仮想的に実験を行っていたので、宇宙線で効率が上がらなければその方法を検討したい）

また、シンチレータを同時に二個通過したときに時間分解能が向上することを確かめ、最終的にはシンチレータを細かくしていったときに時間分解能がどの程度になるかを見積もる。

3 geant 関連の進捗状況

現在、geant4 およびそれを使うための c++ などの勉強を <http://geant4.kek.jp/~iwai/kitasato/> を頼りにやっている。講義資料は説明が少ないので、他のサイトなどで調べながら理解を進めている。サンプルコードは充実していて、C 言語から必要なことを広く学べる。C++ の復習を終えて、やっと geant4 を動かすところまでできた。

4 geant4 などの install について

4.1 install の前に

今回インストールした環境は、opensuse11.2 で gcc は 4.4-4.2。

入れるソースは

- Geant4 が 9.3(patch-02)(最新)。
- Root は 5.26/00(最新の一個前)。
- CLHEP は 2.0.4.7(最新の一個前)。
- DAWN は 3.88a(最新の一個前)。
- DAVID は 1.36a(最新)。

opensuse は OpenGL の代わりに Mesa というドライバーを使う。Mesa-devel は必要なので入れておく。

geant4 はコンパイルに時間がかかる (1 時間ぐらい?) ので、CLHEP を入れたら、次に geant4 のコンパイルを始めて、その間に ROOT とか DAWN を入れるとよいかも。

4.2 install の仕方

基本的に公式サイトの install の解説を読めば問題ない。以下は install のメモなので、流れの確認や、二度めのインストールのときに役立つかも。

4.3 CLHEP

インストールは./configure --prefix <path>
make
make install
だけ。

CLHEP は物理定数とかベクトルとか定義されてて便利。これを使うには、適当なヘッダファイル (include/CLHEP/Units/SystemOfUnits.h とか) と CLHEP ライブラリ (lib/libCLHEP.so とか) を読み込めばよい。

ただし、namespace が CLHEP となっているので、
using namespace CLHEP
と一行初めに書いておく。

4.4 GEANT4

data ファイルは geant4.9.3.p02/data ってフォルダに入れる。あとは、./Configure -build
でいろいろ聞かれるが、

Please, specify where CLHEP is installed: /CLHEP

Enable building of the X11 OpenGL visualization driver? [n] y [ENTER]

Enable building of the X11 DAWN visualization driver? [n] y [ENTER]

Enable building of the X11 RayTracer visualization driver? [n] y [ENTER]

これ以外はデフォルトで問題ない。あとは公式サイトガイドに従うのみ。

4.5 ROOT

ソースは pro が安定版。環境変数は (bash).bashrc に

ソースコード 1 環境変数

```
export ROOTSYS=<path>/root
```

```
export PATH=$ROOTSYS/bin:$PATH
```

```
export LD_LIBRARY_PATH=$ROOTSYS/lib:$LD_LIBRARY_PATH
```

と書いておいて、bash を再起動か source .bashrc

で、

```
./configure
```

```
make
```

```
make install
```

で終わり。root

で起動、.q で終了。C のソースが生で実行できるのがおもしろいと思った。

4.6 DAWN

デフォルトでは tk,tcl が入っていないかったので、入れた。後は、公式どおりにインストール。

```
./configure
```

このあと色々聞かれるが、こんな感じで答えた。

ソースコード 2 DAWN

```
Input C++ compiler name (e.g. g++, no default).
```

```
: g++
```

```
Input optimization option (e.g. -O2, no default)
```

```
: -O2 (ハイフンオーツー)
```

```
Input Tcl/Tk window shell name
```

```
: wish
```

```
Input include directory of Xlib "with -I":
```

```
: -I/usr/include/X11
```

```
Input lib directory of Xlib "with -L":
```

```
: -L/usr/lib
```

```
Input the first include directory of OpenGL "with -I":
```

```
: -I/usr/include
```

```
Input the second include directory of OpenGL "with -I":
```

```
: -I/usr/include/GL
```

```
Input lib directory of OpenGL "with -L":
```

```
: -L/usr/lib
```

```
Select OpenGL/Mesa or old Masa (0: OpenGL/Mesa , 1: Old Mesa (ver.3.0)): 0
```

```
Input directory to install executable files
```

```
: /home/ryuki/geant4/DAWN/bin
```

```
Select a system (0: Defalut , 1: AIX, 2: IRIX, 3: Solaris , 4: RedHat Linux): 0
```

```
Do you use remote-visualization function? (y/n, no default)
```

```
: n
```

後は

```
make
```

```
make install
```

で終わり。

4.7 DAVID

KEK の公式に install の方法が書いてあるのでその通りに。Makefile がいくつか種類があるが、たぶん

```
make -f Makefile.GNU_g++
```

でいいはず。test がうまくいったら、Makefile の install 先 (INSTALL_DIR) を書き換えて、

make install

で完了。

5 細かいこと

5.1 geant4 の勉強中にはまったこと

上で紹介したサイトでソースに問題があり、さっきかなり深くハマったので紹介する。

geant4 の makefile はいくつか必要事項を書いて、makefile の雛形を呼んで終わりだが、
G4WORKDIR = .

と書いてあり、これを

```
G4WORKDIR = $(shell pwd)
```

と直すべきである。

問題は、相対パスは展開されずに、そのまま実行時リンクのパスとして渡されてしまうことにある。この場合、実行ファイルの場所が変わったときに動的ライブラリの場所がわからなくなってしまう。

ちなみに、リンクがうまくいってるかは、

```
ldd main
```

とやれば見える。実行時リンクの張り方は (例えば /usr/local/lib にパスを通すとする)、

```
.bashrc に export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
```

と書くか

リンク時にオプションで `-Wl,-rpath /usr/local/lib`

とする。

5.2 shellsript ではまったこと

ソースコード 3 shellsript

```
if [ $1 = $2 ]; then
  echo ‘‘hello world’’
fi
```

について、if の後の `[]` は `test` というコマンドのエイリアスだが、括弧の前後の空白が正しくないとダメ。