

Курс практических занятий

Введение в современные программные средства для моделирования физических процессов и анализа

Введение

Цель этого курса - ознакомление с современными программными средствами для моделирования физических процессов, происходящих с частицами. Обычно речь будет идти о частицах высокой энергии, от MeV (мегаэлектронвольт, 10^6 eV) до TeV (тераэлектронвольт, 10^{12} eV), но иногда нужно моделировать движение частиц и более низкой энергии, до 1 eV.

1. Операционные системы и оболочки

Основной операционной системой в крупных научных центрах, таких как ЦЕРН, является UNIX, а в последние лет восемь это Linux. ЦЕРН поддерживает разновидность Scientific Linux CERN, сокращенно SLC (<http://linux.cern.ch>). Компьютеры кластера SLC общего пользования в CERN называются lxplus. Они поддерживают файловую систему AFS. Тома этой системы видны со всех компьютеров lxplus, а также с большинства desktop компьютеров, установленных в CERN. Дистанционный логин в компьютеры CERN извне разрешен только через SSH (ssh, sftp, scp).

Некоторые эксперименты в CERN поддерживают WINDOWS в качестве одной из платформ. Кроме того, при решении мелких задач каждый может выбирать себе наиболее удобную операционную систему. SSH доступ к UNIX компьютерам из WINDOWS возможен через PuTTY. Скачать соответствующие программы (PuTTY и PSFTP) можно здесь: <http://www.chiark.greenend.org.uk/~sgtatham/putty>

В интернете краткое введение в UNIX можно найти здесь: <http://www.dkpw.co.uk/rute/rute.html> или здесь: <http://rute.2038bug.com/index.html.gz> (можно также сделать поиск "rute.html" в Google)

Важным инструментом при разработке программ является Make. Его краткое описание можно найти в документе, упомянутом выше, а более подробное здесь: <http://www.gnu.org/software/make/manual/>

В UNIX существует несколько оболочек, наиболее распространенные – это c-shell, или csh (чаще ее расширение tcsh) и b-shell, или sh (в Scientific Linux и других разновидностях Red Hat Linux, таких как Fedora, sh указывает на bash). Эксперименты, как правило, поддерживают обе эти оболочки, но одна из них включается по умолчанию. Специалисты иногда советуют писать scripts (если они не совсем простые) на b-shell. Некоторые соображения по этому поводу можно прочитать здесь: <http://www.grymoire.com/Unix/index.html>

Упражнение 1.1

Написать script на sh и csh, который копирует все файлы из директории на один уровень выше в текущую директорию. Сделать это, написав цикл в явном виде.

Упражнение 1.2

Написать Makefile, компилирующий программу myprog.cc (executable назвать myprog). Компиляцию делать в два этапа: сначала собственно компиляция, создание объектного файла myprog.o (опция компилятора -c), затем сборка (linking). Убедиться, что, если сделать второй раз make, программа не перекомпилируется. Добавить зависимость от include файла myname.inc и убедиться, что если его изменить, программа перекомпилируется. Использовать стандартные

переменные Make \$@ и \$<

2. ROOT

ROOT служит для анализа данных и их графического представления. Он написан на с++ и имеет в своем составе CINT – интерпретатор с++. Site здесь: <http://root.cern.ch>

В CERN ROOT установлен на AFS: /afs/cern.ch/sw/lcg/external/root. В нем имеется папка tutorials с многочисленными примерами использования ROOT. Часто бывает проще выбрать один из примеров и модифицировать его с помощью reference manual (описание классов и методов), чем писать с нуля свой макрос, пользуясь user's manual, который имеет довольно большой объем.

Чтобы установить ROOT на компьютере, нужно сделать следующие шаги:

1. Скачать ROOT source tree с указанного выше WEB site, file называется root_vX.XX.XX.tar.gz, где X.XX.XX – версия ROOT. Этот файл положить в директорию /swdisk/ROOT, где swdisk – partition или путь туда, где устанавливается software. Лучше пользоваться production version, а не development.
2. Перейти в эту директорию, развернуть ROOT tarball: tar xvfz root_vX.XX.XX.tar.gz
В результате будет создана директория root. Переименовать ее в X.XX.XX:
mv root X.XX.XX Это нужно, поскольку, возможно, потом придется поставить другую версию ROOT.
3. Перейти в директорию ROOT. Установить environment variable ROOTSYS:
setenv ROOTSYS /swdisk/ROOT/X.XX.XX (для shell csh, tcsh) или
export ROOTSYS=/swdisk/ROOT/X.XX.XX (для shell sh, bash). Тем самым реализуется environment-driven installation method из инструкции, которая находится в /README/INSTALL.
4. Сконфигурировать ROOT: configure linux –enable-roofit Здесь первый аргумент – вид системы, он подходит для таких систем как Fedora, slc4, slc5 на Intel компьютерах. Вторым аргументом (перед словом enable 2 минуса) установит дополнительный пакет RooFit, который потом понадобится. Полный список возможных аргументов можно получить если сделать
configure –help.
5. make
6. make install
7. Создать environment для работы ROOT:
setenv LD_LIBRARY_PATH \${ROOTSYS}/lib (csh,tcsh) или
export LD_LIBRARY_PATH=\${ROOTSYS}/lib (sh,bash)
setenv PATH \${PATH}:\${ROOTSYS}/bin (csh,tcsh) или
export PATH=\${PATH}:\${ROOTSYS}/bin (sh.bash)
после этого еще может понадобиться сделать команду rehash
8. С этим environment ROOT будет запускаться из любого места командой root. Установку переменных ROOTSYS, LD_LIBRARY_PATH, PATH (пункты 3,7) желательно занести в стартовый файл .cshrc (.shrc).

Упражнение 2.1

Написать ROOT макрос (файл с extension .C), который строит graph функции f(x), где x задан точками (10,15,20,30,40,60,70), f точками (200,250,300,400,500,700,780), с ошибками, равными квадратному корню из f. Воспользоваться ROOT /tutorials и reference manual.

3. Event Generators

Взаимодействия частиц, которые характерны для эксперимента и много раз повторяются в нем, часто моделируются специальными программами, называемыми Event Generators. Такие программы позволяют точнее моделировать некоторые наиболее интересные (при этом часто редкие) процессы.

Среди Event Generators различают General Purpose Generators и более узкоспециализированные Matrix Element Generators. Первые способны моделировать взаимодействие до уровня наблюдаемых частиц, а вторые моделируют только первичное взаимодействие, описываемое определенным матричным элементом. В конечном состоянии Matrix Element Generators могут быть глюоны и кварки – ненаблюдаемые частицы. С такими частицами в дальнейшем происходит адронизация, в результате которой возникают наблюдаемые частицы. Дело в том, что адронизация не описывается в теории возмущений. Поэтому не существует точных формул, описывающих это явление, а только полуфеноменологические модели, реализованные в генераторах первого типа (General Purpose). Это означает, что Matrix Element Generators могут использоваться только совместно с одним из General Purpose Generators.

К General Purpose Generators относятся PYTHIA6, HERWIG, PYTHIA8 и HERWIG++. Первые два написаны на FORTRAN, а последние два являются их развитием, реализованном на с++ с использованием Object Oriented technologies. В генераторах PYTHIA для адронизации используется модель рвущихся струн, а в генераторах HERWIG – модель кластеризации. Ожидается, что в ближайшие год - два, после интенсивного тестирования с++ генераторов, FORTRAN предшественники перейдут в разряд устаревших.

Matrix Element Generators, в силу отсутствия сложных частей, отвечающих за адронизацию партонов, гораздо более многочисленны. Наиболее известные генераторы такого типа – ALPGEN, MADGRAPH, MCatNLO.

Самые известные репозитории генераторов – HepForge (<http://www.hepforge.org/projects>) и GENSER в CERN (<http://lcgapp.cern.ch/project/simu/generator/>). Задача проекта GENSER – поддерживать генераторы и вспомогательные пакеты (такие как LHAPDF – пакет Parton Density Functions) для экспериментов LHC. Для многих FORTRAN генераторов GENSER предоставляет build system, стараясь приблизиться к GNU стандартам. Пакеты GENSER установлены в CERN на AFS: /afs/cern.ch/sw/lcg/external/MCGenerators

Наиболее распространенный в настоящее время способ хранения выходных данных генераторов (на уровне наблюдаемых частиц) – это HepMC event record (<http://lcgapp.cern.ch/project/simu/HepMC/>). Пакет HepMC установлен в CERN на AFS: /afs/cern.ch/sw/lcg/external/HepMC.

Упражнение 3.1

Установить HepMC на компьютере (production version);

Взять дистрибутив PYTHIA6 из GENSER:

/afs/cern.ch/sw/lcg/external/MCGenerators/distribution/pythia6-413-src.tgz;

скомпилировать PYTHIA6;

модифицировать пример main78 из /examples: выводить на экран поперечный импульс Pt Z-бозона в каждом событии, ввести нижнюю границу на эту величину при моделировании событий и убедиться, что это работает (нет событий с Pt меньше границы), и что сечение процесса уменьшается после введения границы;

сделать вывод в файл и написать ROOT макрос, который читает этот файл и строит гистограмму Pt.

Упражнение 3.2

Включить гистограммирование с помощью ROOT в executable упражнения 3.1. Необходимые

header файлы, библиотеки и опции компилятора посмотреть в /ex3.2. Построить гистограмму Pt Z-бозона и нарисовать ее.

Упражнение 3.3

Повторить упражнения 3.1 и 3.2 для PYTHIA8.

4. GEANT4

Эта программа (так называемый toolkit, то есть коды, из которых можно выбирать нужный пользователю набор) служит для моделирования прохождения частиц через вещество, в том числе в присутствии электромагнитных полей. В ней также моделируются взаимодействия частиц, но лишь те, вероятность которых не очень мала. Web site здесь:

<http://geant4.web.cern.ch/geant4/>

Как обычно, для знакомства с этой программой удобно пользоваться примерами. Папка /examples в дистрибутиве содержит их довольно много и имеет разветвленную структуру.

Посмотрим на пример N02 в /novice. В папке src находятся несколько файлов с кодами.

В файле *DetectorConstruction.cc определяется геометрия установки.

В файле *PhysicsList.cc определяются нужные частицы и процессы. Определение процессов с частицами, не имеющими сильного взаимодействия (как примеры N02, N03) имеет обычно довольно простую структуру. Если же нужно работать с адронами не очень низкой энергии, то нужно использовать стандартные листы из библиотек GEANT4.

В файле *SD.cc определяются действия, которые должны осуществляться, когда частица попадает в части установки, определенные как чувствительные детекторы.

Упражнение 4.1

Если не установлен, то установить вспомогательный пакет CLHEP

<http://proj-clhep.web.cern.ch/proj-clhep/>

В CERN CLHEP установлен на AFS в external (полный путь см. выше).

Скачать дистрибутив GEANT4 с Web site и растарить его;

В папке /ex4.1 можно найти файл configure4.9.1 (подставить правильную версию в название). Его нужно отредактировать, подставив правильные пути к CLHEP и GEANT4, затем сделать source ./configure4.9.1;

Войти в папку /source и запустить make. Компиляция занимает почти час;

Скопировать в свою рабочую папку пример N02, войти в эту папку; взять в ex4.1 файл configure, отредактировать его (правильные пути к CLHEP и GEANT4), source ./configure, make;

После этого программу в batch можно запустить так: ./bin/Linux-g++/exampleN02 run1.mac

А в интерактивной моде, с графикой, так: ./bin/Linux-g++/exampleN02. Выйти из программы в интерактивной моде можно командой exit

Упражнение 4.2

Скачать программу vtmlview, например здесь: <http://www.km.kongsberg.com/sim>

Binary желательно установить в /usr/local/bin. Программа старая, при запуске она будет требовать старую версию libstdc++. Нужно установить compat версию gcc (yum list | grep gcc; yum install ...). Если она недостаточно старая, то можно сделать symlink, например: /usr/lib/libstdc++-libc6.1-1.so.2 -> libstdc++-3-libc6.2-2-2.10.0.so

В Windows файлы VRML можно смотреть программой Deep Exploration.

Заменить в файле vis.mac DOWNFILE на VRML1FILE в команде /vis/open;

Запустить GEANT4 binary без параметра (при этом используется файл vis.mac);

Образуются файл g4_00.wrl (при повторном запуске g4_01.wrl и т.д.). Его можно смотреть

программой vtmlview;

В интерактивной моде можно посмотреть какие команды имеются в данной программе (они имеют древесную структуру). Для этого нужно использовать команду help. Help имеет соответствующую древесную структуру. Кроме стандартных команд GEANT4 есть команды, существующие только в данном example, они находятся под директорией N02;

Найти в help команду, включающую магнитное поле, и убедиться, что траектория частицы искривляется;

Упражнение 4.3

Откомпилировать пример N03;

Вывести в файл полное энергосодержание частицы (электрона) в калориметре и построить гистограмму с помощью ROOT. RMS этой гистограммы можно считать разрешением калориметрического детектора по энергии (более точно фитировать гистограмму распределением Гаусса).

Убедиться, что

1. Разрешение ухудшается при увеличении количества пассивного вещества в калориметре (sampling fluctuations).
2. Разрешение улучшается при увеличении энергии частицы примерно по закону $1/\sqrt{E}$ (выше энергии примерно 1 ГэВ).