

实验模拟与数据分析工具

(Tools for Experimental Simulation and Data Analysis)

2011～2012学年春季课程
课程编号： 312017Z

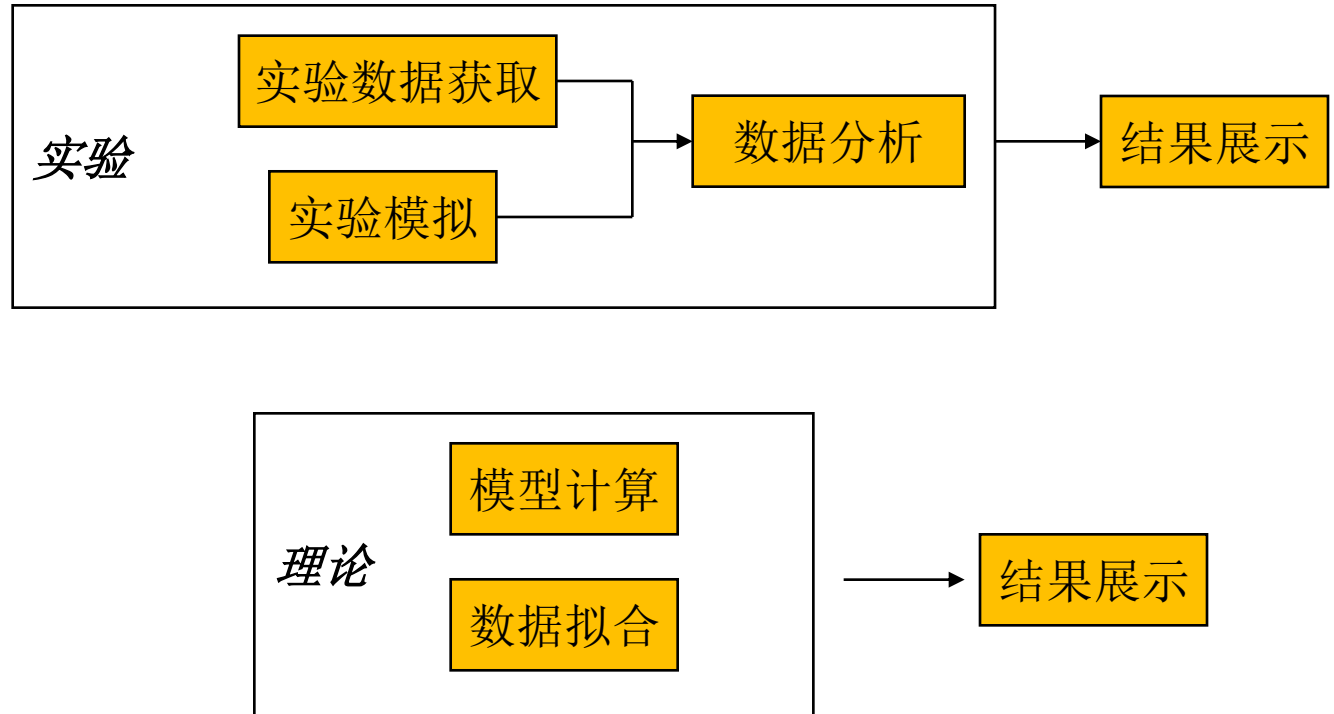
吕晓睿（中国科学院研究生院）

Analysis Tools in Physics Research

2



Geant 4



通过学习对各种物理分析工具的学习，了解和熟悉对物理研究中的分析方法的应用和理解。

教学内容

3

第一章 引言

第二章 实验数据分析工具ROOT

ROOT的内存管理；Hitogram/Ntuples/trees；DATA structure implementation；数据的图形表达；数据拟合工具RooFit；多变量分析工具TMVA。

第三章 物理事件产生器（Event Generator）

物理反应机制模拟；程序流程与设置；Numbering Scheme；实例分析。

第四章 探测器模拟

探测器模拟对象（几何、材料和物理过程等）；面向对象的程序框架和流程；实例分析。

第五章 实验物理软件与分析框架系统

Monte Carlo模拟软件框架；实验数据流系统；数据分析框架。

知识准备与学习材料

4

Basics:

- ❑ Linux: <http://www.linux.org>
- ❑ C++: <http://www.cplusplus.com>
- ❑ Monte Carlo: [http://en.wikipedia.org/wiki/Monte Carlo method](http://en.wikipedia.org/wiki/Monte_Carlo_method)
- ❑ Wikipedia: <http://en.wikipedia.org>
- ❑ Dictionary: <http://www.thefreedictionary.com>
- ❑ 

Word-Wide-Web (WWW):

- 课程网站
- “The ROOT Users Guide”, Rene Brun et al., <http://root.cern.ch>
- PYTHIA, <http://home.thep.lu.se/~torbjorn/Pythia.html>
- GEANT Manuel, <http://cern.ch/geant4>

References:

- 朱永生, 《实验数据多元统计分析》, 科学出版社, 2009

实验模拟与数据分析工具

教学计划

5

- 课堂讲授（周二N408；周五N408）+上机操作（机房）
- 上机时间：上课时间
暂定日期：2/24； 3/2； 3/16； 3/30； 4/13；
课下自学练习：**非常重要**
如有需要，可安排额外上机时间
课堂讲义 网上资源 操作练习
- 期末考试：4月27日
- 作业与答疑：
 - 课程网站
 - E-mail: cp_gucas@163.com
 - Office Hour: 周三下午15:00-17:00
高能所主楼A518 (Tel: 88236386)
- 考核：作业 + 期末考试

实验模拟与数据分析工具

教学服务器

6

- 提供Linux环境用于课上演示，上机操作练习和完成作业
- IP: 210.76.201.75 (校园网内登录，IP随时可能会变动，注意课程网站上的通知。)
- O/S: Scientific Linux 4
- 用户名: 名简拼+姓全拼+学号后三位 (xr1vXXX)
- 密码: cp12S@gucas 登陆后请修改密码 (passwd)
- 登陆方式: ssh + scp; 提供Windows下登陆软件 (到课程网站下载): XManager(推荐)
- 课上所有示例均在/users/comphys12S/cp中
- 提交作业到目录:
/users/comphys12S/HomeWork/\${num}/\${用户名}

实验模拟与数据分析工具

第一章：物理分析软件ROOT

吕晓睿（中国科学院研究生院）



ROOT

...creates the file, the structure
TFile f1("tree.root", "reco");
TTree *t1 = new TTree("t1", "A simple tree with
11.Branch("pt", &pt, "p");
t1.Branch("py", &py, "p");

[Search](#)[Login](#)[Home](#)[What's New](#)[About](#)[Screenshots](#)[Download](#)[Documentation](#)[Support](#)[Forum](#)[Developers](#)

Screenshots

Get a taste of ROOT's capabilities by sampling some screenshots.



Download

Go ahead and download the latest build of ROOT.



Documentation

Get the inside scoop on how to fully utilize ROOT. Also, search the Reference Guide, the HowTo's and the user forums.

What's New

- March 2, 2009, 16:23
Development release
5.23/02
- February 26, 2009, 16:22
Patch release 5.22/00a
- December 18, 2008,
18:14
Production release

Development release 5.23/02

development release

The development release of ROOT 5.23/02 is now available.

The SVN tag for this version is v5-23-02.

[Read more](#)

Patch release 5.22/00a

What is ROOT

3

- An Object-Oriented Data Analysis Framework aimed at solving the data analysis challenges of (high energy) particle physics.
- 一个基于面向对象设计的数据分析软件框架

Rene Brun



Fons Rademakers



navigation

- [Main page](#)
- [Contents](#)

[article](#)

[discussion](#)

[edit this page](#)

[history](#)

ROOT

From Wikipedia, the free encyclopedia

For other uses of "root", see [root \(disambiguation\)](#).

ROOT is an [object-oriented program](#) and [library](#) developed by [CERN](#). It was originally designed for [particle physics data analysis](#) and contains several features specific to this field, but it is also commonly ^{[citation needed](#)} used in other applications such as [astronomy](#) and [data mining](#).

History of ROOT

4

- November 29, 1995: ROOT Version 0.04 Released
- February 5, 1996: ROOT Version 0.05 Released
- January 14, 1997: ROOT Version 0.90/12 Released
- April 25, 1997: ROOT Version 1.00 Released
- October 30, 1998: New ROOT version 2.00/13.
- June 8, 2001: Official Release of ROOT Version 3.01
- ...
- December 15, 2010, **Pro, version 5.28/00**
- ...
- December 2, 2011, Production release 5.32/00

release early and release often

Getting Information

5

- ❑ **ROOT website:** <http://root.cern.ch>

You can find ROOT user references and guides there.

- ❑ **the ROOT mailing list:** roottalk@root.cern.ch

- ❑ **subscribe to the mailing list:**

<http://root.cern.ch/root/Registration.phtml>

- ❑ **forums and archives:**

<http://root.cern.ch/root/roottalk/AboutRootTalk.html>

<http://root.cern.ch/phpBB2/>

Framework的作用

6

framework: programming inside a framework is a little like living in a city. Plumbing, electricity, telephone, and transportation are services provided by the city. In your house, you have interfaces to the services such as light switches, electrical outlets, and telephones. The details, for example, the routing algorithm of the phone switching system, are transparent to you as the user. You do not care; you are only interested in using the phone to communicate with your collaborators to solve your domain specific problems.

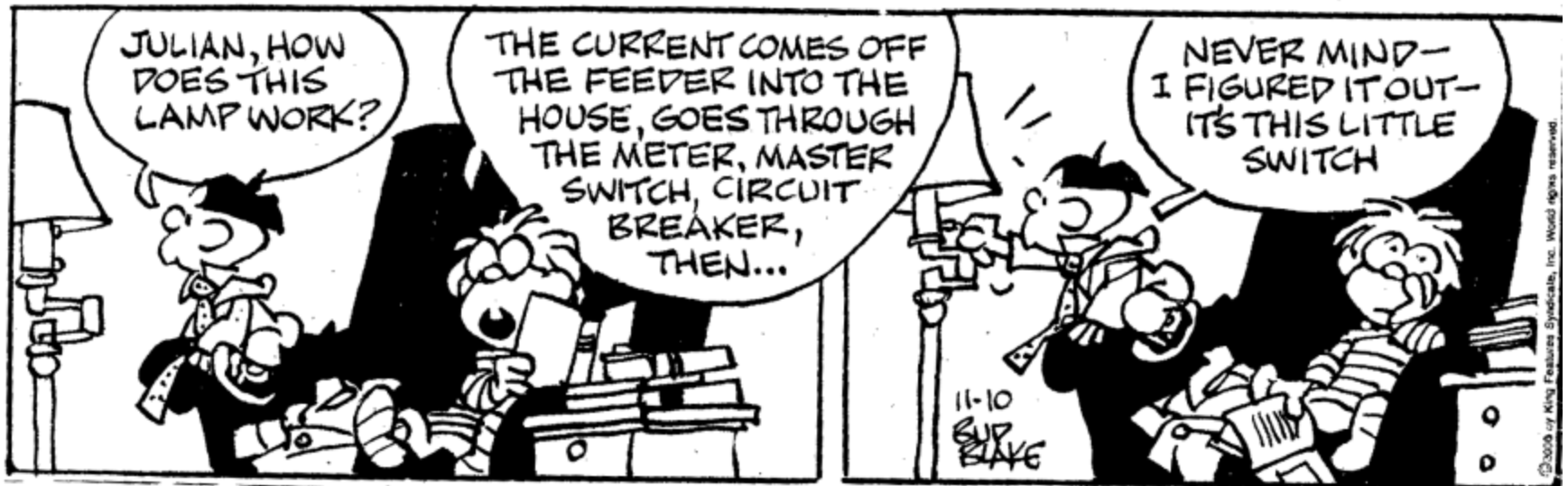
- ❑ **Less code to write**
- ❑ **More reliable and robust code**
- ❑ **More consistent and modular code**
- ❑ **More focus on areas of expertise**

A Framework

7

provides utilities and services.

TIGER By Bud Blake



物理分析软件ROOT

Object Orientated Programming (OOP) in ROOT

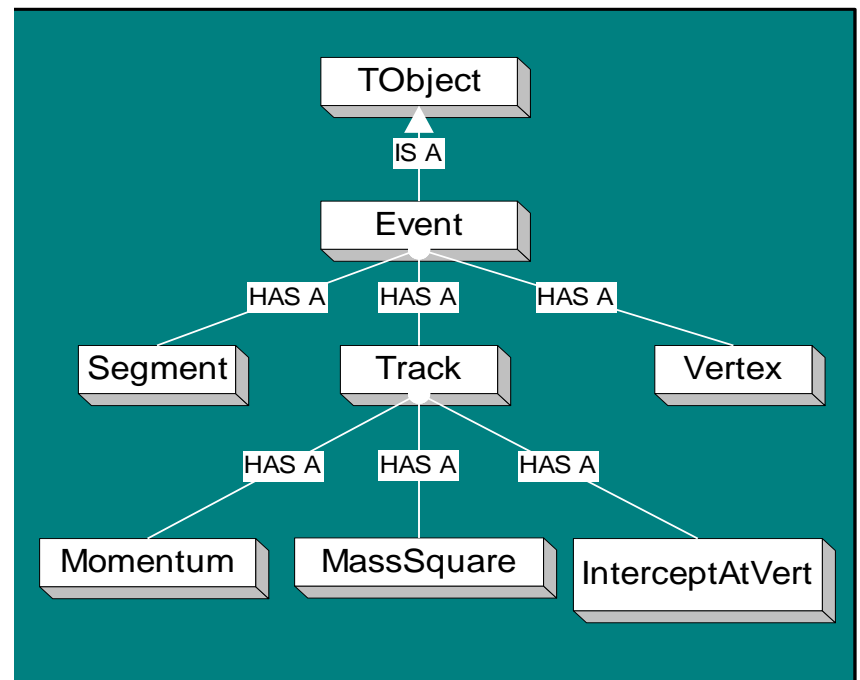
8

- ❑ Encapsulation enforces data abstraction and increases opportunity for reuse.
- ❑ Sub classing and inheritance make it possible to extend and modify objects.
- ❑ Class hierarchies and containment hierarchies provide a flexible mechanism for modeling real-world objects and the relationships among them.
- ❑ Complexity is reduced because there is little growth of the global state, the state is contained within each object, rather than scattered through the program in the form of global variables.
- ❑ Objects may come and go, but the basic structure of the program remains relatively static, increases opportunity for reuse of design.

Object Oriented Concepts

9

- Class: the description of a “thing” in the system
 - Object: instance of a class
 - Methods: functions for a class
-
- Members: a “has a” relationship to the class.
 - Inheritance: an “is a” relationship to the class.



ROOT's Services/Utilities

10

- Histogramming and Fitting
- Graphics (2D, 3D)
- I/O to file or socket: specialized for histograms, Ntuples (Trees)
- Collection Classes and Run Time Type Identification
- User Interface
 - GUI: Browsers, Panels, Tree Viewer
 - Command Line interface: C++ interpreter CINT
 - Script Processor (C++ compiled \Leftrightarrow C++ interpreted)

More ROOT Packages

11

The packages provided by ROOT include those for

- ⑩ [histogramming](#) and [graphing](#) to visualize and analyze [distributions](#) and [functions](#),
- ⑩ [curve fitting](#) (regression analysis) and minimization of [functionals](#),
- ⑩ [statistics](#) tools used for [data analysis](#),
- ⑩ [matrix](#) algebra,
- ⑩ [four-vector](#) computations, as used in [high energy physics](#),
- ⑩ standard [mathematical functions](#),
- ⑩ [multivariate data analysis](#), e.g. using [neural networks](#),
- ⑩ *image manipulation*, used e.g. to analyze [astronomical](#) pictures,
- ⑩ access to *distributed data* (in the context of the [Grid](#)),
- ⑩ [distributed computing](#), to parallelize [data analyses](#),
- ⑩ [persistence](#) and [serialization](#) of objects, which can cope with changes in class definitions of persistent data,
- ⑩ access to [databases](#),
- ⑩ [3D visualizations](#) (geometry)
- ⑩ creating files in various *graphics formats*, like [PostScript](#), [JPEG](#), [SVG](#),
- ⑩ interfacing [Python](#) and [Ruby](#) code in both directions,
- ⑩ interfacing [Monte Carlo event generators](#).

物理分析软件ROOT

Installing ROOT

12

- ❑ ROOT website at: <http://root.cern.ch/root/Availability.html>
- ❑ Support almost all the platforms: UNIX-like system, Windows, Mac OS ...; GCC or Visual C++ compiler
- ❑ The compiled binaries or the source codes available to install
- ❑ Compilation Steps:

```
% tar zxvf root_v2.25.xx.source.tar.gz
% export ROOTSYS=<path>/root
% cd root
% ./configure --help
% ./configure <target>
% gmake
% gmake install
```

- ❑ More information in README/INSTALL or website.
- ❑ You can also try to install in Windows.

Running ROOT

13

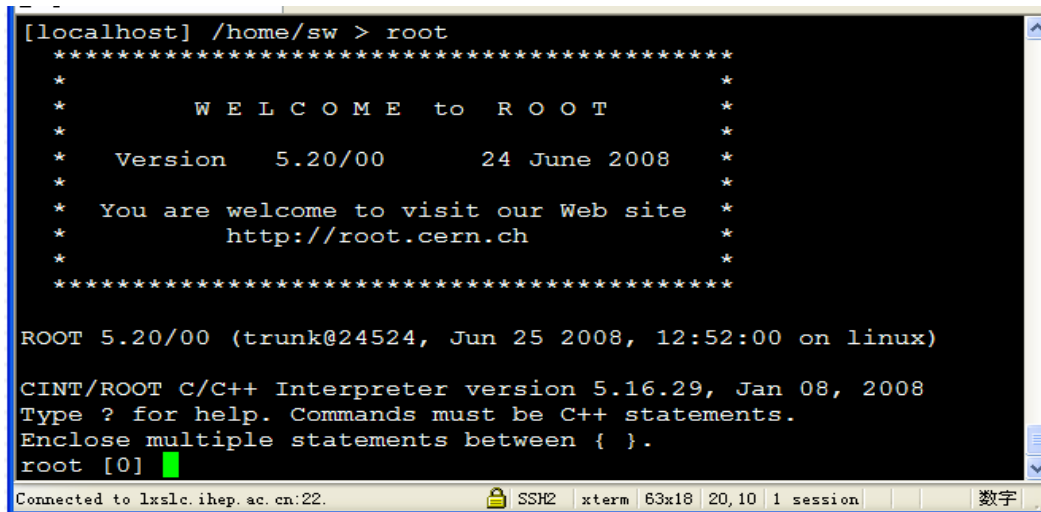
□ set environment variables:

```
> export ROOTSYS=/home/sw/root  
> export PATH=$PATH:$ROOTSYS/bin  
> export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ROOTSYS/lib
```

OR

```
> source /home/sw/root/bin/thisroot.sh
```

□ put the above commands in the login (e.g., *.bashrc*) file

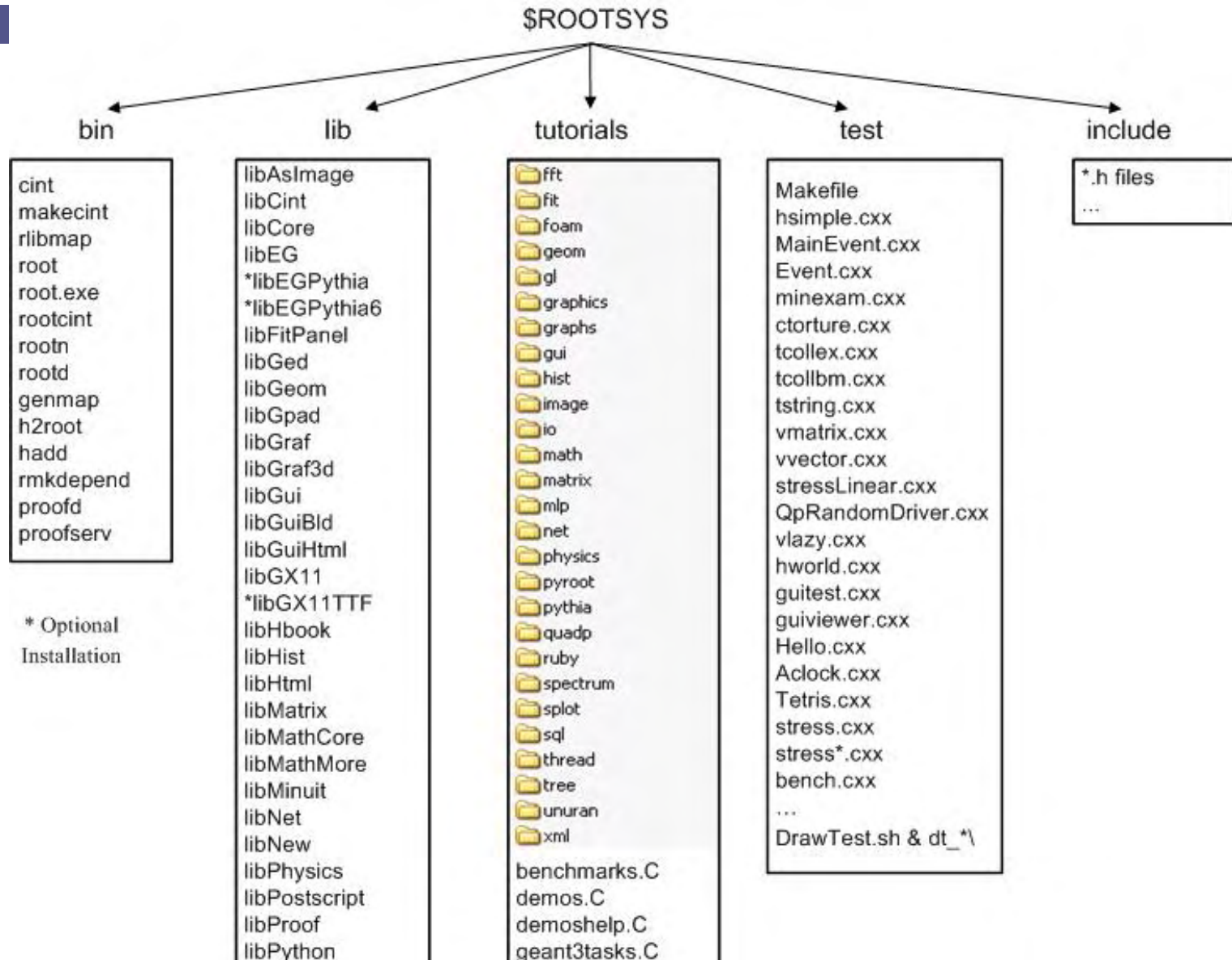


```
[localhost] /home/sw > root  
*****  
*                               *  
*      W E L C O M E  t o  R O O T      *  
*                               *  
*  Version   5.20/00      24 June 2008  *  
*                               *  
*  You are welcome to visit our Web site *  
*      http://root.cern.ch              *  
*                               *  
*****  
  
ROOT 5.20/00 (trunk@24524, Jun 25 2008, 12:52:00 on linux)  
  
CINT/ROOT C/C++ Interpreter version 5.16.29, Jan 08, 2008  
Type ? for help. Commands must be C++ statements.  
Enclose multiple statements between { }.  
root [0]
```

新软件ROOT

The Framework Organization

14



\$ROOTSYS/bin

15

<code>root</code>	shows the ROOT splash screen and calls <code>root.exe</code>
<code>root.exe</code>	the executable that <code>root</code> calls, if you use a debugger such as <code>gdb</code> , you will need to run <code>root.exe</code> directly
<code>rootcint</code>	is the utility ROOT uses to create a class dictionary for CINT
<code>rmkdepend</code>	a modified version of <code>makedepend</code> that is used by the ROOT build system
<code>root-config</code>	a script returning the needed compile flags and libraries for projects that compile and link with ROOT

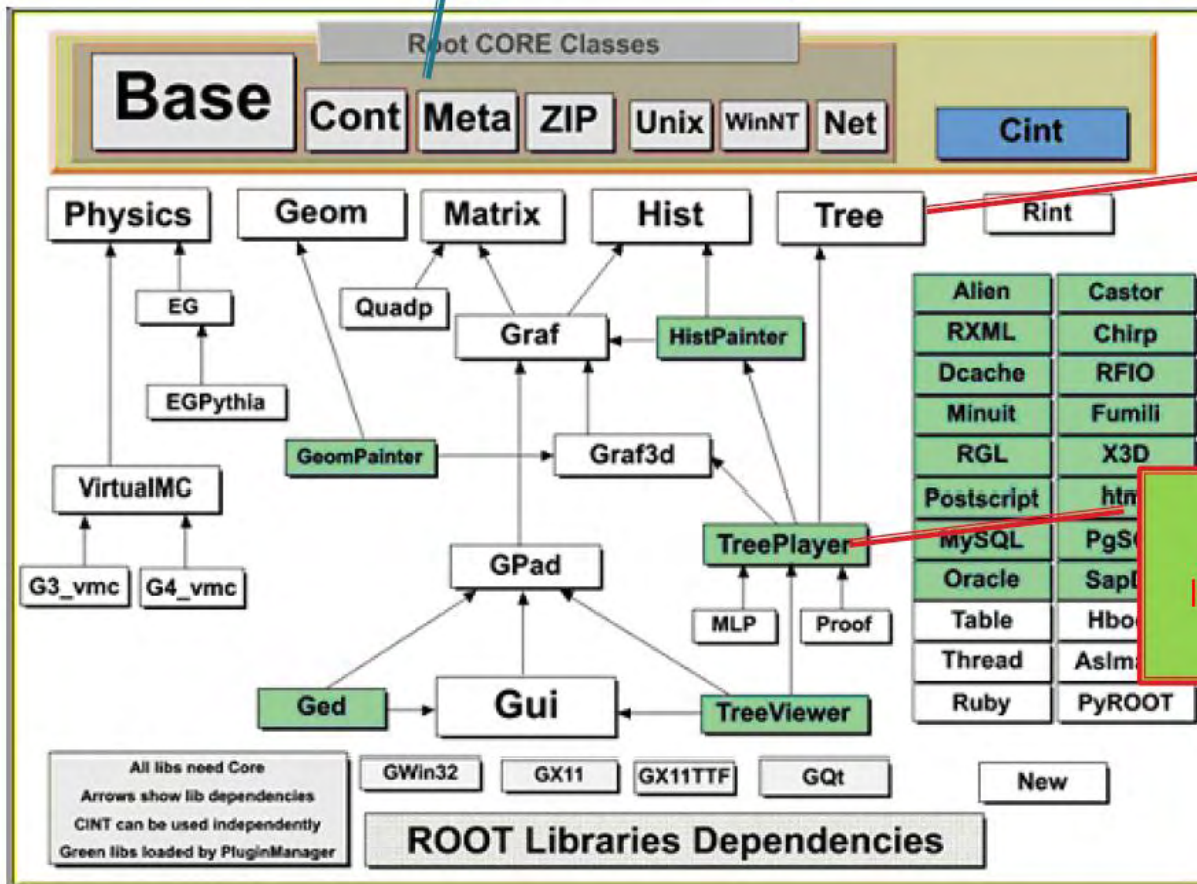
others: `proofd`, `proofserv`, `rootd`...

The Libraries

16

Essentials
Needed in all

Over 350 classes!!!



Need to
be
explicitly
linked

Auto
loaded

Special purpose
libraries: EG,
Physics...

件ROOT

ROOT tutorials and test

17

- \$ROOTSYS/tutorials: contains **many example scripts**. They assume some basic knowledge of C++ and ROOT
- \$ROOTSYS/test: a set of examples that represent all areas of the framework and **a gold mine**

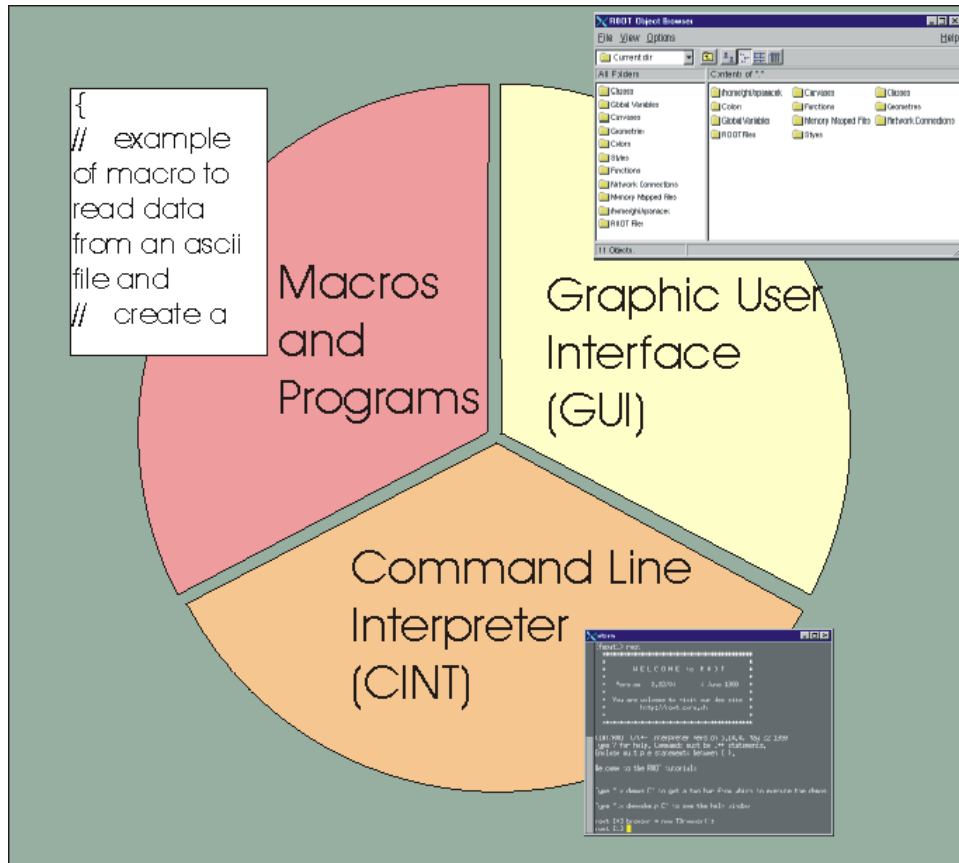
More Information on ROOT

18

- Class Reference Guide:
 - <http://root.cern.ch/root/html/ClassIndex.html>
- User's Guide
 - <http://root.cern.ch/root/doc/RootDoc.html>
- ROOT Tutorials
 - <http://root.cern.ch/root/Tutorials.html>
- HowTos
 - <http://root.cern.ch/root/HowTo.html>
- Example Applications
 - <http://root.cern.ch/drupal/content/example-applications>
- Courses
 - <http://root.cern.ch/drupal/content/tutorials-and-courses>

Three User Interfaces

19



- GUI
windows, buttons, menus
- Root Command line
CINT (C++ interpreter)
- Macros, applications,
libraries (C++ compiler and
interpreter)

Start and Quit ROOT

20

- To start ROOT, you can type `root` at the system prompt. This starts up CINT, the ROOT command line C/C++ interpreter, and it gives you the ROOT prompt (`root[0]`).
- To quit ROOT, type `.q` at the ROOT prompt
- It is possible to launch ROOT with some command line options, as shown below:

```
> root -h
Usage: root [-l] [-b] [-n] [-q] [file1.C ... fileN.C]
Options:
-b : run in batch mode without graphics
-n : do not execute logon and logoff macros as
specified in .rootrc
-q : exit after processing command line script files
-l : do not show the image logo (splash screen)
```

Command-Line Interface

21

- a powerful C/C++ interpreter giving you access to all available ROOT classes, global variables, and functions via the command line.
- By typing C++ statements at the prompt, you can create objects, call functions, execute scripts, etc.
- Use up and down arrows to recall commands:
\$HOME/.root_hist
- Use emacs commands to navigate

```
root[] 1+sqrt(9)
(const double)4.000000000000000000e+00
root[] for (int i = 0; i<2; i++) cout << "Hello" << i << endl
Hello 0
Hello 1
```

Command-Line Interface

22

□ different types of commands:

1. CINT commands start with “.”

```
root[] .?//this command will list all the CINT commands  
root[] .L <filename> //load [filename]  
root[] .x <filename> //load and execute [filename]
```

2. SHELL commands start with “.!” for example:

```
root[] .! ls
```

3. C++ commands follow C++ syntax (almost: not standard C++)

```
root[] TBrowser *b = new TBrowser()
```

4. Multi-line Commands: put the multi-line commands in the double curly brackets { }.

Use double-[<Tab>-click to search for the complement](#): to find List of Methods and Parameters

物理分析软件ROOT

CINT Commands

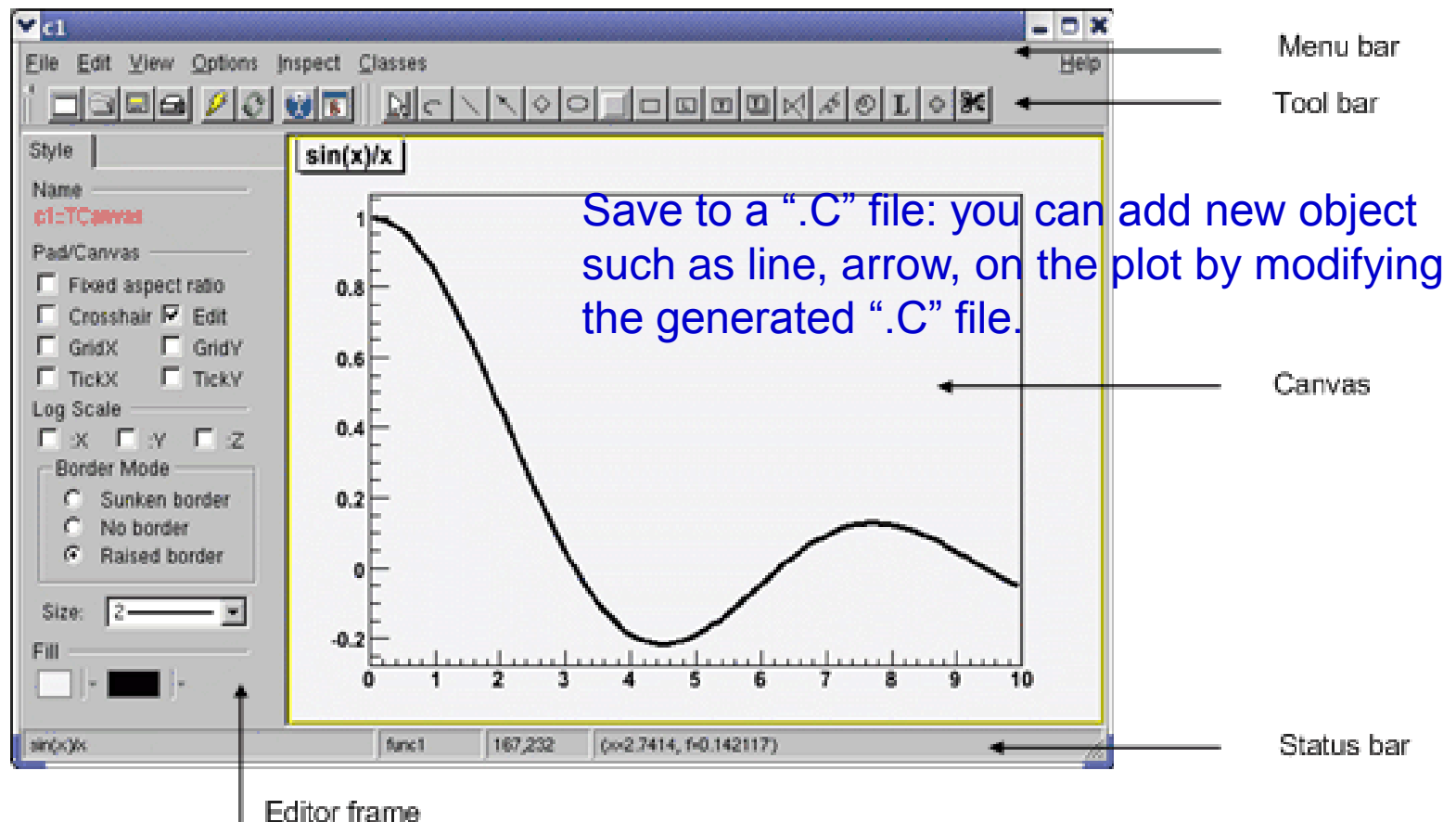
23

- `[expression]` evaluates the expression
 `root[3] 3*4`
 `(int)12`
- `.files` show loaded source files
- `.class [name]` show class definition
- `.g` prints all objects in the root session
- `.ls` ls on current directory
- `.pwd` list the current directory, canvas and style.
- Unix-like I/O redirection (semicolon ; is required before > or >>)
- Use ? to get help on all “raw” interpreter commands
- Use @ to abort a multi-line command

GUI Interface

24

```
root[] TF1 f1("func1","sin(x)/x",0,10)
root[] f1.Draw()
<TCanvas::MakeDefCanvas>: created default TCanvas with name c1
```



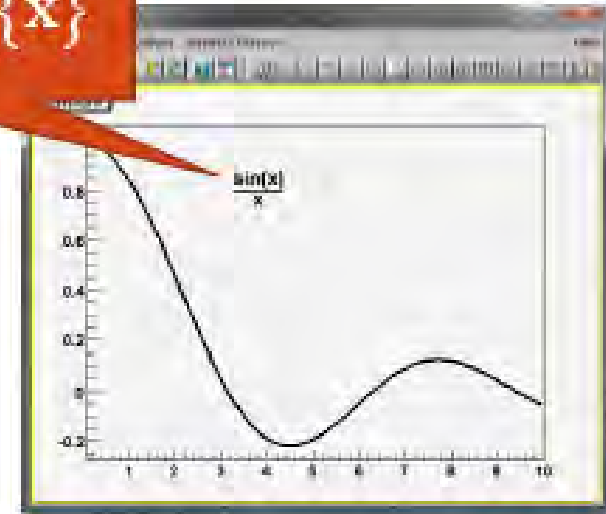
Fast Generate Your Macro Code

25

Save the plot to c1.C file:

$\#frac{\sin(x)}{x}$

```
{
//=====Macro generated from canvas: c1/c1
//===== (Fri Mar 21 08:10:57 2008) by ROOT version5.18/00
TCanvas *c1 = new TCanvas("c1", "c1",423,36,699,534);
c1->Range(-1.119108,-0.4477822,11.24013,1.22563);
c1->SetBorderSize(2);
c1 >SetFrameFillColor(0);
c1->TF1 *func1 = new TF1("func1","sin(x)/x",0.1,10);
func1->SetFillColor(19);
func1->SetFillStyle(0);
func1 >SetLineWidth(3);
func1->func1->Draw("");
TPaveText *pt = new TPaveText(0.01,0.9401777,0.1371429,0.995,"b1NDC");
pt->SetName("title");
t >S tB d Si (2) pt->SetBorderSize(2);
pt->SetFillColor(19);
TText *text = pt->AddText("sin(x)/x");
pt->Draw();
tex = new
TLatex(0.261097,0.768812,"#frac{sin(x)}{x}") ;
tex->SetLineWidth(2) ;
tex->Draw() ;
c1->Modified();
c1->cd();
c1->SetSelected(c1);
c1->ToggleToolBar();
}
```



Add these three lines to put a math formula on the plot and re-run the macro c1.C

物理分析软件ROOT

Macros Applications and Libraries

26

- write a C++ code file, where ROOT class can be directly used
- execute the C++ code file without compilation (take it as Macros) -- **convenient!!!**

```
> root myMacro.C  
root[] .x myMacro.C  
> root -b -q 'myMacro.C(3)' > myMacro.log  
> root -b -q 'myMacro.C("text")' > myMacro.log  
> root -b -q "myMacro.C(\"text\")" > myMacro.log
```

- Use ACLiC to build a shared library
- Based on ROOT libraries to produce your own libraries or executables

Environment Setup

27

- The behavior of a ROOT session can be configured with the options in the *rootrc* file. At start-up, ROOT looks for a *.rootrc* file in the following order:

<code>./rootrc</code>	<code>//local directory</code>
<code>\$HOME/.rootrc</code>	<code>//user directory</code>
<code>\$ROOTSYS/etc/system.rootrc</code>	<code>//global ROOT directory</code>

- to see current settings, you can do:

```
root[] gEnv->Print()
```

- Default Logon and Logoff Scripts:

in local directory:

`rootalias.C` (only loaded when root starts up)

`rootlogon.C` (executed when root starts up)

`rootlogoff.C` (executed when root ends)

in `$HOME` directory: `.rootlogon.C`

物理分析软件ROOT

examples: check the relevant files `$ROOTSYS/tutorials`

Start ROOT from \$ROOTSYS/Tutorials

28

- execute root under \$ROOTSYS/tutorials, the following information appears when root is up:

```
> root
...
Welcome to the ROOT tutorials
Type ".x demos.C" to get a toolbar from which to execute the demos
Type ".x demoshelp.C" to see the help window
==> Many tutorials use the file hsimple.root produced by hsimple.C
==> It is recommended to execute hsimple.C before any other script
root[0].x demos.C
```

- Also you can try: *root <filename.C>* to check the separate macro outputs.
- “*grep*” those macros to find what you need
- More tutorials at <http://root.cern.ch/root/html/index.html>

grep使用

29

- 一种强大的文本搜索工具，它能使用简单字符串/正则表达式搜索文本，可以用于查找例程寻求帮助。如：
- 查找fit目录中使用TF1类的例程
`grep "TF1" $ROOTSYS/tutorials/fit/*.C`
- 查找所有目录中使用TF1类的例程
`grep -r "TF1" $ROOTSYS/tutorials/fit`
- 查找roofit例程中的gauss函数使用（“-w”表示查找单词）
`grep -rw "gauss" $ROOTSYS/tutorials/roofit`
- 查找忽略大小写查找
`grep -rwi "gauss" $ROOTSYS/tutorials/roofit`
- 以某些首字母开头字符串查找
`grep -r "gauss.*" $ROOTSYS/tutorials/roofit`

ref: <http://man.chinaunix.net/newsoft/grep/open.htm>

The ROOT Script Processor

30

□ Un-named Scripts:script1.C

```
{  
#include <iostream.h>  
cout << " Hello" << endl;  
float x = 3.; float y = 5.; int i = 101;  
cout <<" x = "<<x<<" y = "<<y<<" i = "<<i<< endl;  
}
```

root[] .x script1.C <enter>

□ Named Scripts:script2.C

```
#include <iostream.h>  
int run(int j=10)  
{  
cout << " Hello" << endl;  
float x = 3.; float y = 5.; int i= j;  
cout <<" x = "<< x <<" y = "<< y <<" i = "<< i << endl;  
return 0;  
}
```

The ROOT Script Processor

31

```
root [] .L script2.C
root [] .func
...
script2.C2:7 0 public: int run(int j=10);
root [] run(<tab>
int run(int j = 10)
root [] run()
    Hello
    x = 3 y = 5 i = 10
(int)0
root [] run(1)
    Hello
    x = 3 y = 5 i = 1
(int)0
```

- Change the function name to the script prefix name:

`int run(int j=10)` ➔ `int script2(int j=10)`

then you can execute the macro via:

```
root[] .x script2.C(8)
```

Object definition in CINT Script

32

```
// file name: draw1.C
void draw(){
    TF1 f("f","sin(x)",0,20);
    f.Draw();
}
```

local variable/object is destroyed when the function exits.

modified ones:

```
// file name: draw2.C
void draw(){
    TF1 *f=new TF1("f","sin(x)",0,20);
    f->Draw();
}
```

```
// file name: draw3.C
{
    TF1 f("f","sin(x)",0,20);
    f.Draw();
}
```

what the difference?

课程小结

33

- ROOT框架简介
 - 如何运行/退出ROOT程序
 - ROOT界面使用介绍
 - ROOT图形使用
 - 课后阅读：
 - ▣ ROOT User Guide第一章和第二章
- To be familiar with the jargon words,
To see what ROOT can do,
To feel ROOT's power.

课下练习

34

- Linux一些简单使用
 - ▣ 设置ROOTSYS, PATH, LD_LIBRARY_PATH
 - ▣ .bashrc作用
 - ▣ vim/emcas使用
 - ▣ 简单Linux命令: cd, ls, cp, mv, rm, ...
- 结合user guide运行tutorial内的程序
- 建议
 - ▣ 借（下载）本入门（电子）书看
 - ▣ 多用多练多问
 - ▣ baidu/google