

Introducción a ROOT

Clase 0: Uso Básico

Fernando Quiñonez PhD.
Universidad Industrial de Santander

2016 Junio 22

Overview

Instalación

Bases

ROOT como calculadora de tienda

ROOT Scripts

ROOT/C++ para High Performance Computing HPC

Configuración

Apuntadores Globales

Instalación

Instalar ROOT:

Seguir los pasos descritos en:

Wiki OPENGATE Collaboration

Varias maneras de iniciar ROOT:

```
root
```

```
root -l
```

```
root archivo.C
```

```
root -b archivo.C
```

```
root -b -q myMacro.C > myMacro.log
```

```
root -b -q 'myMacro.C(3)' > myMacro.log
```

```
root -b -q 'myMacro.C("text")' > myMacro.log
```

```
root -b -q myMacro.so > myMacro.log
```

root -h

```
quinonez@Aspire-V5-471: ~  
File Edit View Search Terminal Help  
quinonez@Aspire-V5-471:~$ root -h  
Usage: root [-l] [-b] [-n] [-q] [dir] [[file:]data.root] [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 macro files  
  -l : do not show splash screen  
  -x : exit on exception  
dir : if dir is a valid directory cd to it before executing  
  
  -?      : print usage  
  -h      : print usage  
  --help  : print usage  
  -config : print ./configure options  
  -memstat : run with memory usage monitoring  
  
quinonez@Aspire-V5-471:~$
```

Salir de ROOT

```

File Edit View Search Terminal Help
quinonez@Aspire-V5-471:~$ root
Couldn't find font "-adobe-helvetica-medium-r-*-*-10-*-*-*-*-*iso8859-1",
trying "fixed". Please fix your system so helvetica can be found,
this font typically is in the rpm (or pkg equivalent) package
XFree86-[75,100]dpi-fonts or fonts-xorg-[75,100]dpi.

*****
*
*      W E L C O M E   t o   R O O T      *
*
*   Version   5.34/36      5 April 2016   *
*
* You are welcome to visit our Web site *
*      http://root.cern.ch              *
*
*****

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on linuxx8664gcc)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0] .q
quinonez@Aspire-V5-471:~$ █

```

Cuando root se queda *trabado* hay que salir usando múltiplos de 3 de la letra q:

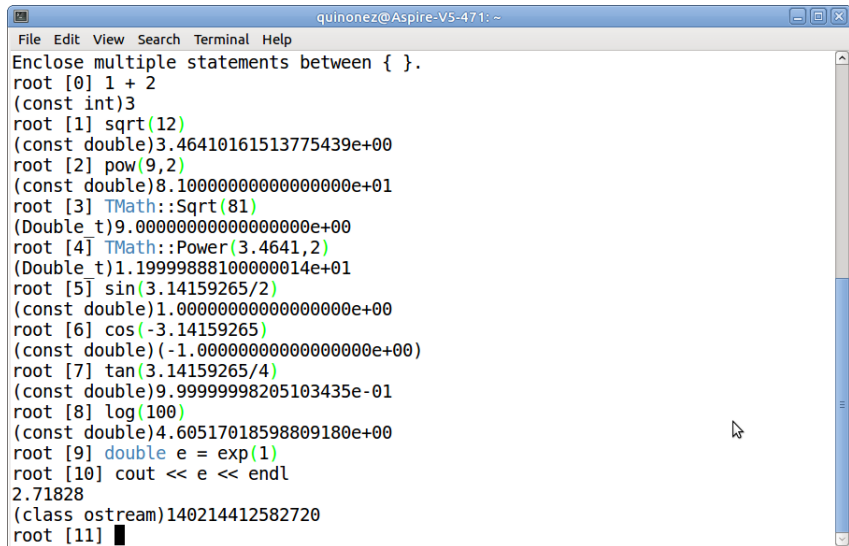
.q

.qqq

.qqqqqq

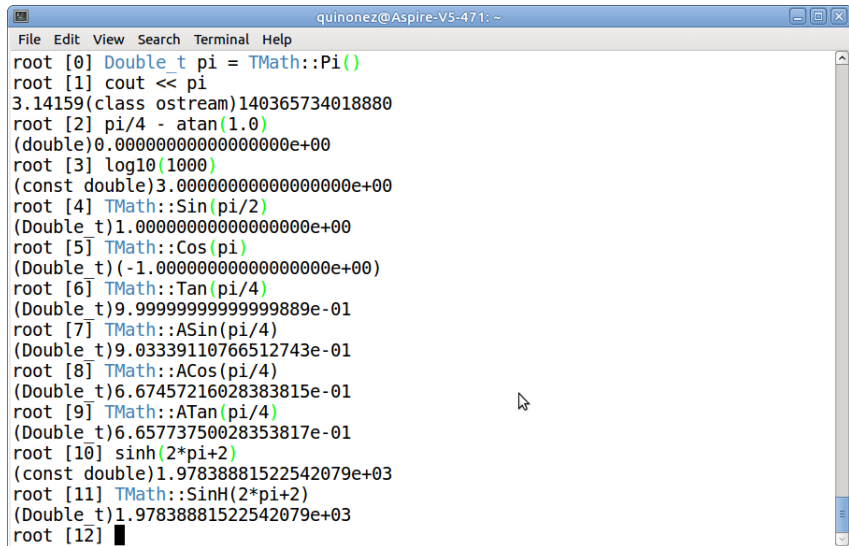
.qqqqqqqqqq

ROOT como calculadora



```
quinonez@Aspire-V5-471: ~  
File Edit View Search Terminal Help  
Enclose multiple statements between { }.  
root [0] 1 + 2  
(const int)3  
root [1] sqrt(12)  
(const double)3.46410161513775439e+00  
root [2] pow(9,2)  
(const double)8.10000000000000000e+01  
root [3] TMath::Sqrt(81)  
(Double_t)9.00000000000000000e+00  
root [4] TMath::Power(3.4641,2)  
(Double_t)1.19999888100000014e+01  
root [5] sin(3.14159265/2)  
(const double)1.00000000000000000e+00  
root [6] cos(-3.14159265)  
(const double)(-1.00000000000000000e+00)  
root [7] tan(3.14159265/4)  
(const double)9.99999998205103435e-01  
root [8] log(100)  
(const double)4.60517018598809180e+00  
root [9] double e = exp(1)  
root [10] cout << e << endl  
2.71828  
(class ostream)140214412582720  
root [11] █
```

ROOT como calculadora



```
quinonez@Aspire-V5-471: ~  
File Edit View Search Terminal Help  
root [0] Double_t pi = TMath::Pi()  
root [1] cout << pi  
3.14159(class ostream)140365734018880  
root [2] pi/4 - atan(1.0)  
(double)0.0000000000000000e+00  
root [3] log10(1000)  
(const double)3.0000000000000000e+00  
root [4] TMath::Sin(pi/2)  
(Double_t)1.0000000000000000e+00  
root [5] TMath::Cos(pi)  
(Double_t)(-1.0000000000000000e+00)  
root [6] TMath::Tan(pi/4)  
(Double_t)9.9999999999999989e-01  
root [7] TMath::ASin(pi/4)  
(Double_t)9.03339110766512743e-01  
root [8] TMath::ACos(pi/4)  
(Double_t)6.67457216028383815e-01  
root [9] TMath::ATan(pi/4)  
(Double_t)6.65773750028353817e-01  
root [10] sinh(2*pi+2)  
(const double)1.97838881522542079e+03  
root [11] TMath::SinH(2*pi+2)  
(Double_t)1.97838881522542079e+03  
root [12] █
```


ROOT Scripts

1. NN (*no named*).

No hay funciones definidas dentro del script. Simplemente el contenido del script se encierra en un par de corchetes (alcance de bloque de C++) .

2. Con nombre (*named*).

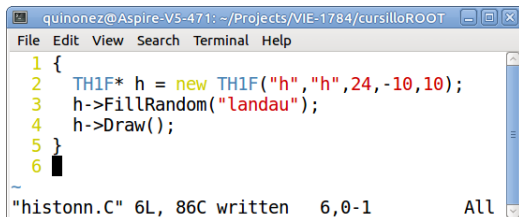
Con la condición de que el nombre de la función principal dentro del script se llame igual que el archivo de extensión .C.

3. ACLIC

Para ser compilados con ACLIC (deben tener nombre y los headers)

Scripts sin nombre

Por ejemplo, se tiene un script sin nombre llamado histo.C, con el siguiente contenido:



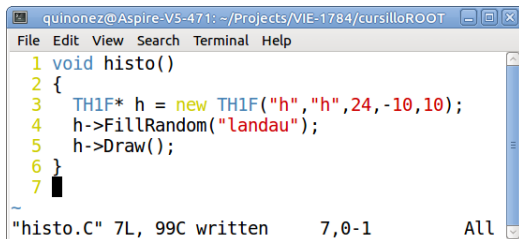
```
quinonez@Aspire-V5-471: ~/Projects/VIE-1784/cursilloROOT
File Edit View Search Terminal Help
1 {
2   TH1F* h = new TH1F("h","h",24,-10,10);
3   h->FillRandom("landau");
4   h->Draw();
5 }
6 █
~
"histonn.C" 6L, 86C written 6,0-1 All
```

Hay “dos” formas de ejecutar el script.

1. `root histonn.C`
2. Dentro de root hacer:
`.x histonn.C`

Scripts con nombre

Por ejemplo, se tiene un script con nombre llamado histo.C, con el siguiente contenido:



```
quinonez@Aspire-V5-471: ~/Projects/VIE-1784/cursilloROOT
File Edit View Search Terminal Help
1 void histo()
2 {
3     TH1F* h = new TH1F("h", "h", 24, -10, 10);
4     h->FillRandom("landau");
5     h->Draw();
6 }
7
~
"histo.C" 7L, 99C written      7,0-1      All
```

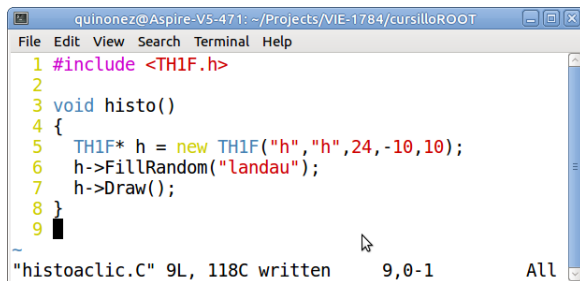
Hay “tres” formas de ejecutar el script.

1. `root histo.C`
2. Dentro de root hacer:
`.x histo.C`
3. Dentro de root hacer:
`.L histo.C`
`histo()`

Scripts compilados con ACLiC

ACLiC: Atomic Compiler of Libraries for CINT.

CINT: C/C++ Interpreter. Por ejemplo, se tiene un script con nombre llamado histoaclic.C, con el siguiente contenido:



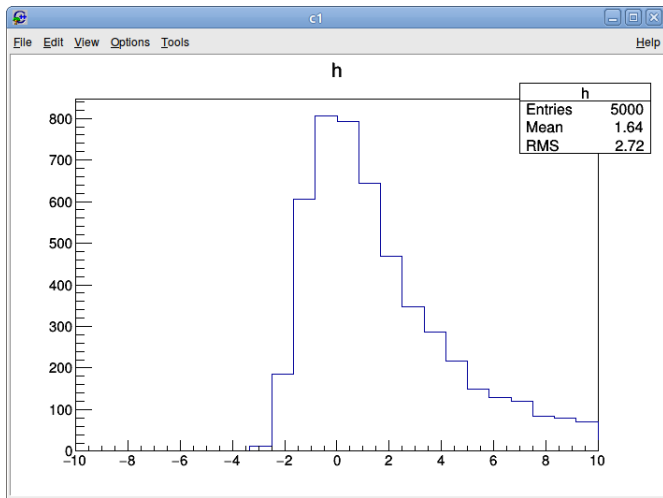
```
quinonez@Aspire-V5-471: ~/Projects/VIE-1784/cursilloROOT
File Edit View Search Terminal Help
1 #include <TH1F.h>
2
3 void histo()
4 {
5     TH1F* h = new TH1F("h","h",24,-10,10);
6     h->FillRandom("landau");
7     h->Draw();
8 }
9
~
"histoaclic.C" 9L, 118C written    9,0-1    All
```

La forma de ejecutar el script es:

```
.L histoaclic.C+
histo()
```

Histograma

En todos los tres casos, se obtendrá:



ROOT/C++ HPC

La idea es integrar ROOT con C++ de la manera acostumbrada en linux mediante el uso de *Makefile's*. Uno (casi siempre) crea un nuevo folder en donde van a quedar los archivos que conforman el *proyecto*.

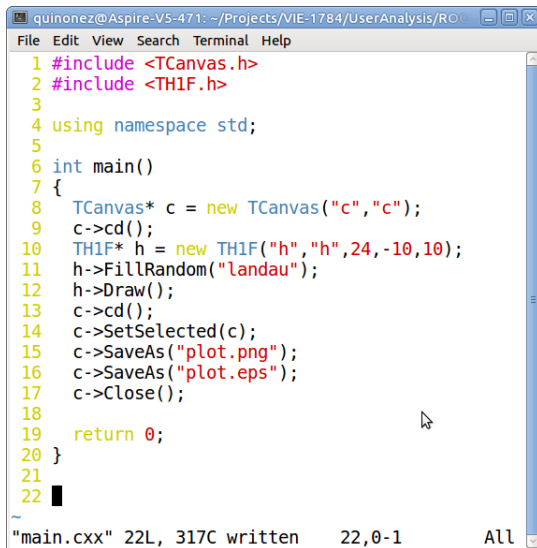
Regla de Oro Sistemas HPC

No se permite que los usuarios pueden usar el servidor X. En castellano, significa que los usuarios no pueden iniciar sesiones gráficas.

Para correr un trabajo que haga el mismo histograma necesitamos un folder con los siguientes archivos:

- ▶ main.cxx
- ▶ mainLinkDef.h
- ▶ Makefile
- ▶ Makefile.arch

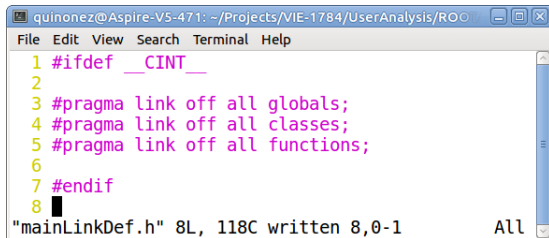
main.cxx



```
1 #include <TCanvas.h>
2 #include <TH1F.h>
3
4 using namespace std;
5
6 int main()
7 {
8     TCanvas* c = new TCanvas("c","c");
9     c->cd();
10    TH1F* h = new TH1F("h","h",24,-10,10);
11    h->FillRandom("landau");
12    h->Draw();
13    c->cd();
14    c->SetSelected(c);
15    c->SaveAs("plot.png");
16    c->SaveAs("plot.eps");
17    c->Close();
18
19    return 0;
20 }
21
22
```

"main.cxx" 22L, 317C written 22,0-1 All

mainLinkDef.h



The screenshot shows a code editor window titled "quinonez@Aspire-V5-471: ~/Projects/VIE-1784/UserAnalysis/ROO". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The code content is as follows:

```
1 #ifdef __CINT__
2
3 #pragma link off all globals;
4 #pragma link off all classes;
5 #pragma link off all functions;
6
7 #endif
8
```

At the bottom of the editor, a status bar displays the file path and statistics: "mainLinkDef.h" 8L, 118C written 8,0-1. On the right side of the status bar, the word "All" is visible.

Makefile

```
quinonez@Aspire-V5-471: ~/Projects/VIE-1784/UserAnalysis/RO...  
1  Autor: Fernando Quinonez  
2  # Para trabajar con linux o macintosh, no existe nada mas :D  
3  # Los codigos fuente deben ser .cxx  
4  include Makefile.arch  
5  mains = main.$(SrcSuf) mainDict.$(SrcSuf)  
6  maino = main.$(ObjSuf) mainDict.$(ObjSuf)  
7  mainso = main.$(DllSuf)  
8  main = main$(ExeSuf)  
9  mainlib = $(shell pwd)/$(EVENTSO)  
10 OBJS = $(maino)  
11 PROGRAMS = $(main)  
12 .SUFFIXES: .$(SrcSuf) .$(ObjSuf) .$(DllSuf)  
13 .PHONY:      Aclock Hello Tetris  
14 # Esta parte da las reglas para hacer las librerias dinamicas  
15 all: $(PROGRAMS)  
16 $(mainso): $(maino)  
17 ifeq ($(PLATFORM),macosx) # 1  
18 # We need to make both the .dylib and the .so  
19 $(LD) $(SOFLAGS)@ $(LDFLAGS) $^ $(OutPutOpt) @ $(EXPLINKLIBS)  
20 ifneq ($(subst $(MACOSX_MINOR),,1234),1234) # 2  
21 ifeq ($(MACOSX_MINOR),4) # 3  
22 ln -sf $@ $(subst .$(DllSuf),.so,$@)  
23 endif # 3  
24 endif # 2  
25 else  
26 $(LD) $(SOFLAGS) $(LDFLAGS) $^ $(OutPutOpt) @ $(EXPLINKLIBS)  
27 endif # 1  
28 @echo "$@ done"  
29 # aqui se construye el ejecutable  
30 $(main): $(mainso)  
31 $(LD) $(LDFLAGS) $(maino) $(LIBS) $(OutPutOpt)@  
32 @echo "$@ done"  
33 clean:  
34 @rm -f $(OBJS) $(PROGRAMS) $(mainso) *Dict.*  
35 .SUFFIXES: .$(SrcSuf)  
36 mainDict.$(SrcSuf): mainLinkDef.h  
37 @echo "Generating dictionary $@..."  
38 $(ROOTCINT) -f $@ -c ^  
39 .$(SrcSuf).$(ObjSuf):  
40 $(CXX) $(CXXFLAGS) -c $<  
41 ifeq ($(GCC_MAJOR),4)  
42 ifeq ($(GCC_MINOR),1)  
43 TBenchDict.o: CXXFLAGS += -Wno-strict-aliasing  
44 endif  
45 endif  
46  
"Makefile" 46L, 1358C written 1,1 All
```

Makefile.arch

Es el mismo que viene con la distribución de ROOT. Su ubicación es:

```
$ROOTSYS/etc/Makefile.arch
```

Compilar, enlazar y ejecutar

```
make  
./main
```

Se obtiene el archivo *plot.png*.

Configuración

`$ROOTSYS/etc/system.rootrc`

Apuntadores Globales

```
gROOT->GetListOfClasses()  
gROOT->GetListOfColors()  
gROOT->GetListOfTypes()  
gROOT->GetListOfGlobals()  
gROOT->GetListOfGlobalFunctions()  
gROOT->GetListOfFiles()  
gROOT->GetListOfMappedFiles()  
gROOT->GetListOfSockets()  
gROOT->GetListOfCanvases()  
gROOT->GetListOfStyles()  
gROOT->GetListOfFunctions()  
gROOT->GetListOfSpecials()  
gROOT->GetListOfGeometries()  
gROOT->GetListOfBrowsers()  
gROOT->GetListOfMessageHandlers()
```

Apuntadores Globales

gDirectory

gPad

gFile

gRandom

gStyle

gEnv