ROOT - I

Onunla hayatta kalma != Onda hayatta kalma != Ona rağmen hayatta kalma

- Giriş
  - Nedir ?
  - Neden iyidir ?
- Kullanım
  - Komut vermek
  - Betik yazmak
  - Betiği *.so kitaplığına derlemek
  - Bağımsız derlemek (uygulama yazmak)
  - ROOT GUI' sını kullanmak
  - Kullanıcı uygulamaları için GUI yaratmak
- ROOT' ta Hayatta Kalmak
  - ROOT kullanım klavuzu' nun kullanımı
  - HTML kaynak belgelendirmesinin kullanımı
  - $ROOTSYS/tutorials dizininin etkin işlevi
  - $ROOTSYS/test dizininin etkin işlevi
- Örnek
  - “Kullanım - Komut vermek” kısmındaki örneğin uygulama hali
- Akşam sefası
  - Ertesi günün sabahına hazırlanması beklenen akşamlık ödevler

Giriş

Nedir ?

- **Kitaplıktır (library) :** *daha önceki derslerde geliştirdiğimiz libCizgiVeOk.so* gibi özelleşmiş kitaplıklardan oluşan çok geniş bir kitaplıktır. libCizgiVeOk.so kitaplığında olduğu gibi, tamamen aynı biçimde, kullanıcı programları içinde kullanılabilir. *(büyük projeler için önerilen yöntem)*
  - *GUI geliştirme kitaplığı*
  - *İşaret-aralık (signal-slot) mekanizmasını sağlayan kitaplık*
  - *Veri ayrıştırması (analiz) kitaplığı, v.b.*

- **C/C++ yorumlayıcısıdır (interpreter) :** algoritma geliştirmeyi kolaylaştırmak için, sanki kabukta kabuk komutlarını çalıştırır gibi ya da kabuk betiği yazar gibi C/C++ yazmayı olanaklı kılan bileşenidir
  - Bu yöntem derlemeye göre daha yavas çalışan programlar üretir fakat:
  - Geliştirilen betiklerin *.so kitaplıklarına derlenmesine ve daha hızlı çalışmalarına olanak tanır.

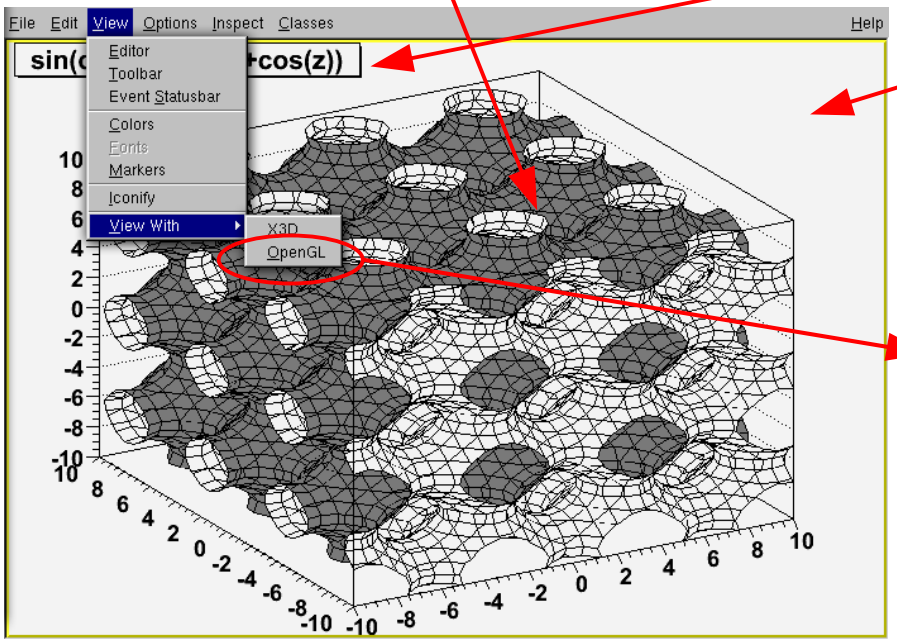
- **Çalışma Ortamıdır *(framework)* :** geliştiricilerin uygulama geliştirmek için ihtiyaç duyacakları seyleri sağlayan bir araçlar toplamıdır
  - Yorumlayıcı, histogram ve veri ayrıştırma (analiz) işlevleri, grafik kullanıcı arayüzü (GUI ve HI) geliştirici, giris/çıkıs (I/O) işlevleri, class kitaplığı, paralel süreç yürütme işlevleri (parallel processing, threads), soket ve ağ haberleşmesi ile ilgi işlevler v.b. işlevler kullanıcının hiçbir sey yapmasına gerek kalmaksızın kullanıma hazırdır.

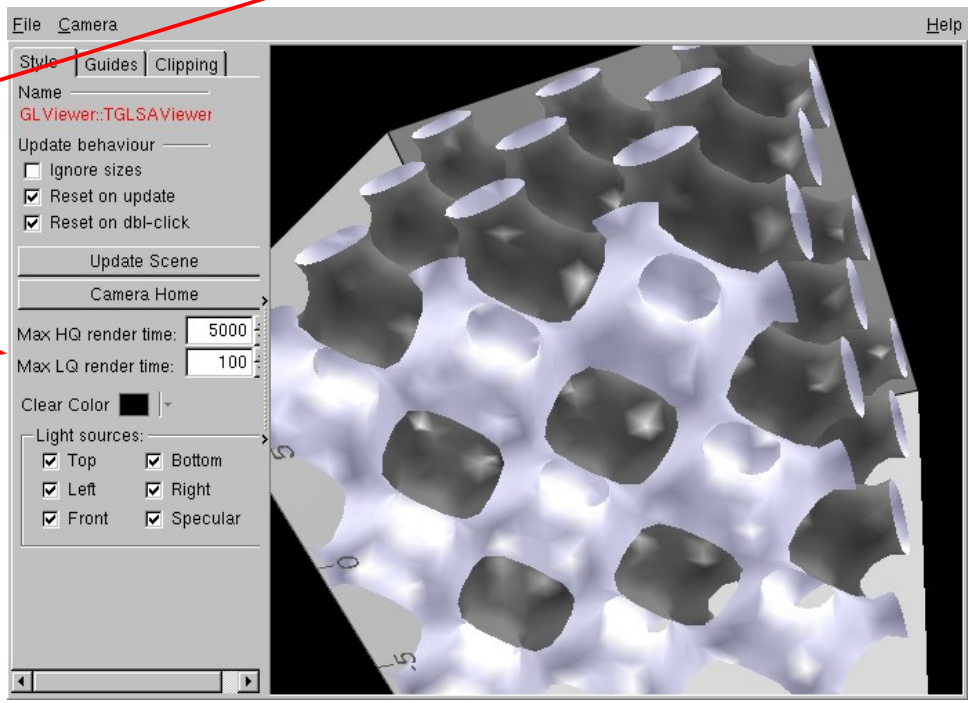
Giriş

Neden iyidir ?

- Çalışma ortamında (framework) olmanın getirileri:
  - Belirli bir işlevsellik için çok daha az kaynak yazmanın gerekliliği
  - Sonuçta geliştirilen programın yüksek bir güvenilirliğe sahip olması
  - Kendi içinde daha tutarlı bir *class* birlikteliği
  - Daha parçalanabilir dolayısı ile daha fazla tekrar kullanılabilir parçalardan oluşan esnek mimari
  - Geliştirici kendi alanına daha fazla yogunlaşabilir

```
root [0] TF3 f1("MerabaDunya","sin(cos(x)+sin(y)+cos(z))",-10,10,-10,10,-10,10)
root [1] f1->Draw()
<TCanvas::MakeDefCanvas>: created default TCanvas with name c1
root [2] MerabaDunya->SetTitle("Bashka bi shey !..")
```

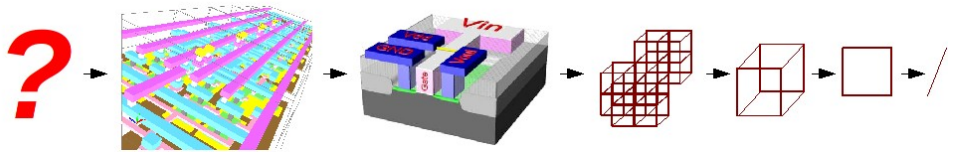




Giriş

Neden iyidir ?

- Çalışma ortamının (framework) nesne yönelimli (object oriented) olmasının getirileri:
  - Nesne yönelimli programlamaya özgü olarak verinin nesne içine gömülü (encapsulation) oluşu, verinin daha soyut düzeylerde ifadesini (abstraction) kolaylaştırdığı için, kitaplığı oluşturan parçaların (class) yeniden kullanılırlığını artırır
  - Class' tan class türetme, varolan nesneleri geliştirmeyi ve değiştirmeyi olanaklı kılar
  - Class' lar arasındaki ast-üst ilişkisi (hiyerarşi), gerçek dünyadaki nesnelerin daha kolay oluşturulmasına imkan verir.

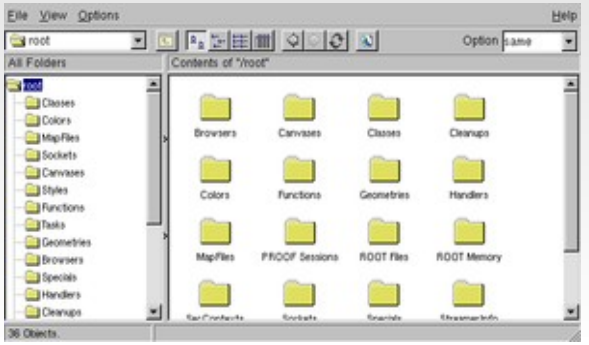


  - Geliştirilen yazılımın karmaşıklığı düşük seviyede kalır, bilgi class' lar içinde düzenli biçimde yerleşiktir; tekdüze bir kaynak içinde küresel değişkenler (globals) üzerinde dağılmış durumda değildir.
  - Yazılımdan nesnelerin çıkarılması ya da yeni nesnelerin eklenmesi, genel mimarii değiştirmez ve böylece belirli bir mimari tekrar tekrar kullanılabilir

Kullanım

Komut vermek

- *Komut girmek, kabukta çalışır gibi*
  - C/C++ ifadeleri ve tüm ROOT class kitaplığı “#include<stdio.h>” ya da “int main()” gibi sıradan programlarda bulunan ifadelere gerek kalmadan girilebilir. Örneğin aşağıdaki komut ile, TBrowser class' ından bir nesne yaratıyor ve adına BenimGuzelGozAticim diyoruz:




```
root [0] TBrowser BenimGuzelGozAticim
root [1] _
```

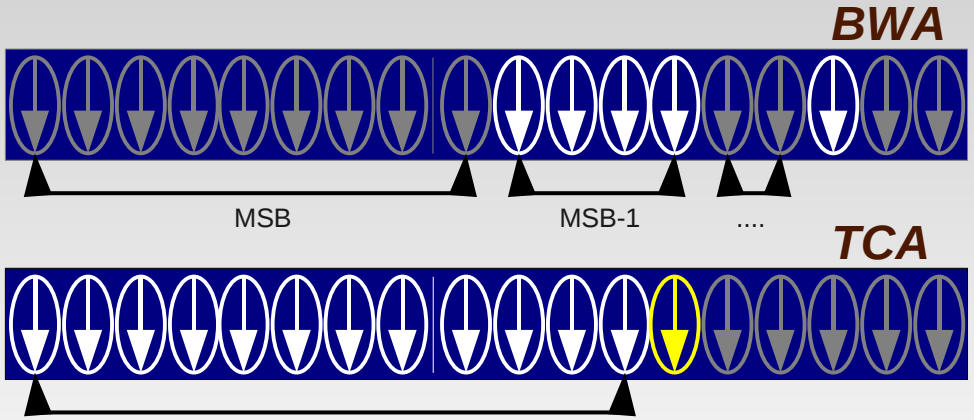
  - Ya da bir döngü:

```
root [0] for (int i=0 ; i<10 ; i++) {
end with '}', '@':abort > printf("%d nin karesi %d \n", i, i*i);
end with '}', '@':abort > }
0 nin karesi 0
1 nin karesi 1
2 nin karesi 4
3 nin karesi 9
4 nin karesi 16
5 nin karesi 25
6 nin karesi 36
7 nin karesi 49
8 nin karesi 64
9 nin karesi 81
root [1] _
```

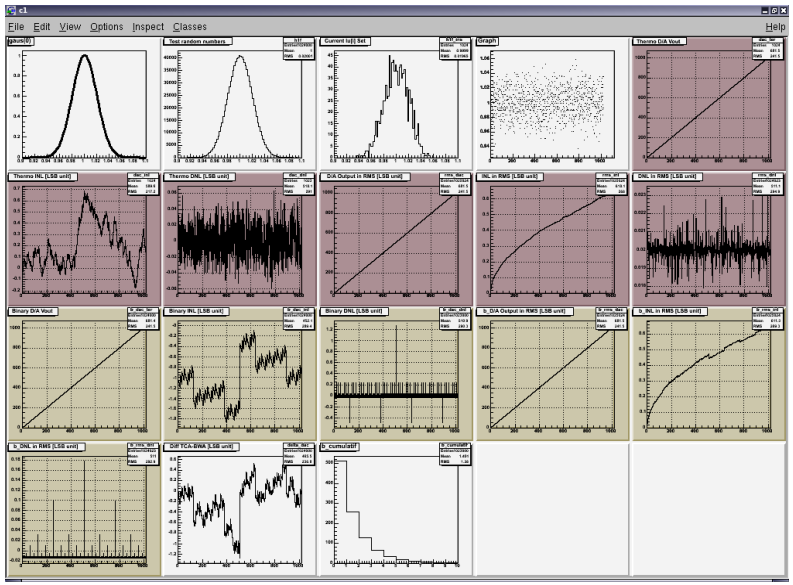
Kullanım

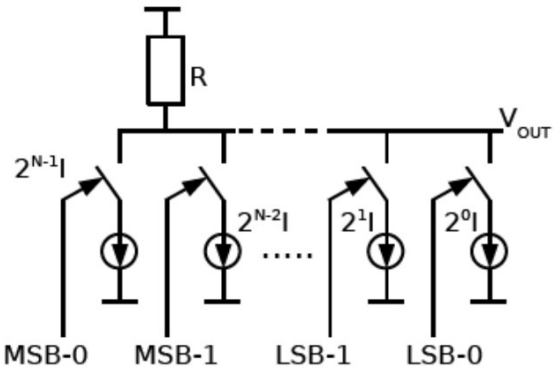
Betik yazmak

- 10-Bit D/A çevirici tasarlanacak
- İki farklı mimari arasında kararsızız
- Hangisi daha uygun belirlemeliyiz ?
- Karşılaştırma gerekli
- Monte Carlo (MC) yapılmalı

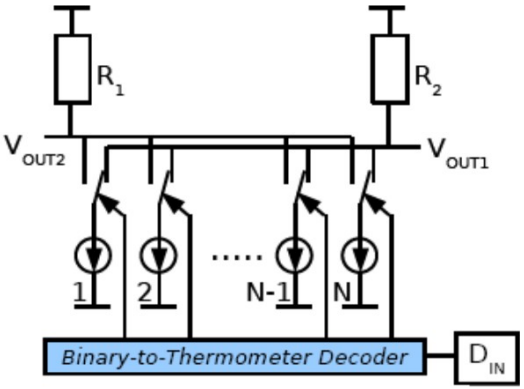


- Çan eğrisinden rastlantısal sayı üret
- Birim standard sapma ile birim akım kaynaklarını belirle
- MC ile her iki mimari için de toplam (integral non-linearity, INL) ve adım başına (differential non-linearity, DNL) doğrusallıktan sapma miktarlarını ve bunların RMS' lerini hesaplamam gerekli





İkili Ağırlıklı (BWA)



Termometre-Kodlu (TCA)

Kullanım

Betik yazmak

File Edit View Search Tools Documents Help

New Open Save Print... Undo Redo Cut Copy Paste Find Replace

mimariKarsilastirici.C | mimariKarsilastirici2.C

```
27 {
28    gROOT->Reset();
29
30    // edit these parameters for other simulations -----------
31    int noofbits=10;
32    int noofiteration4rms=100, nob=100; // nob : no of bins in distro
33    double sigma=0.02, centroid=1.0;
34    const Int_t kUPDATE = 50;
35    bool update=kTRUE;
36    //--------------------------------------------------------
37
38    int nop=pow(2.0, noofbits);
39    int boyut = pow(2, 1*noofbits);
40    double Iu[boyut]; // 2^noofbits current source
41    int lastincremented=0;
42    TCanvas *c1 = new TCanvas("c1","Binary vs Thermometer",1200,850);
43    TPad *pad1 = new TPad("pad1","Unit Current Source",0.0,0.0,1.0,1.0);
44    pad1->Draw();
45    pad1->Divide(5, 4, 0.001, 0.001);
46    pad1->cd(5)->SetFillColor(47);  pad1->cd(5)->SetGrid() ;
47    pad1->cd(6)->SetFillColor(47);  pad1->cd(6)->SetGrid() ;
48    pad1->cd(7)->SetFillColor(47);  pad1->cd(7)->SetGrid() ;
49    pad1->cd(8)->SetFillColor(47);  pad1->cd(8)->SetGrid() ;
50    pad1->cd(9)->SetFillColor(47);  pad1->cd(9)->SetGrid() ;
51    pad1->cd(10)->SetFillColor(47); pad1->cd(10)->SetGrid();
52    pad1->cd(11)->SetFillColor(21); pad1->cd(11)->SetGrid();
53    pad1->cd(12)->SetFillColor(21); pad1->cd(12)->SetGrid();
54    pad1->cd(13)->SetFillColor(21); pad1->cd(13)->SetGrid();
55    pad1->cd(14)->SetFillColor(21); pad1->cd(14)->SetGrid();
56    pad1->cd(15)->SetFillColor(21); pad1->cd(15)->SetGrid();
57    pad1->cd(16)->SetFillColor(21); pad1->cd(16)->SetGrid();
58
59    pad1->cd(1);
60    TH1F *h1f = new TH1F("h1f", "Test random numbers",nob,0.9,1.1);
61    TH1F *h1f_ins = new TH1F("h1f_ins", "Current Iu[i] Set", nob, 0.9, 1.1);
62    TH1F *h1f2= new TH1F("h1f2","Test random numbers COPY",nob,0.9,1.1);
63    TGraph *graph = new TGraph(nop);
64    graph->SetMarkerSize(0);
65    TF1 *gaus = new TF1("gaus","gaus(0)",0.9,1.1);
66    gaus->SetParameters(1, centroid, sigma);
67    gaus->Draw();
68    if (update) c1->Update();
69
70    TH1F *rms_dnl = new TH1F("rms_dnl", "DNL in RMS [LSB unit]",nop-2, 0, nop-2);
71    TH1F *rms_inl = new TH1F("rms_inl", "INL in RMS [LSB unit]",nop, 0, nop);
72    TH1F *rms_dac = new TH1F("rms_dac", "D/A Output in RMS [LSB unit]",nop, 0, nop);
73    TH1F *b_rms_dnl = new TH1F("b_rms_dnl", "b_DNL in RMS [LSB unit]",nop-2, 0, nop-2);
```

Ln 216, Col 21 INS

File Edit View Search Tools Documents Help

New Open Save Print... Undo Redo Cut Copy Paste Find Replace

mimariKarsilastirici.C | mimariKarsilastirici2.C

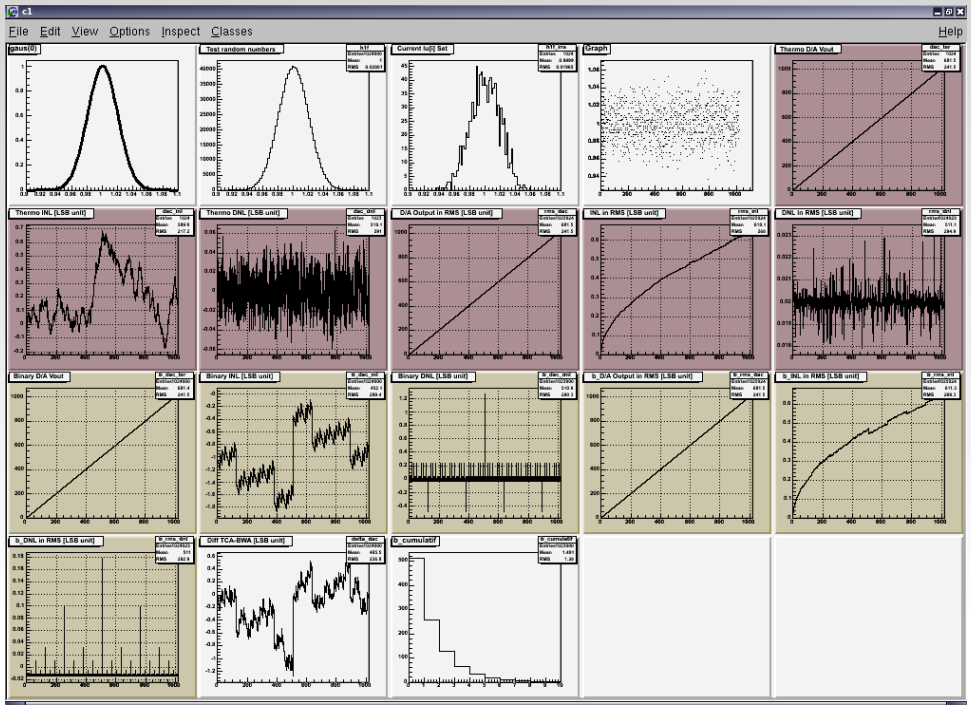
```
182         b_inl[i]=(i+1.0-b_dac[i]);
183         b_rms_inl->SetBinContent(i, b_rms_inl->GetBinContent(i)+b_inl[i]*b_inl[i]);
184         if (i>0 && i<nop) {
185             b_dnl[i-1]=Iu[0]-b_dac[i]+b_dac[i-1];
186             b_rms_dnl->SetBinContent(i-1, b_rms_dnl->GetBinContent(i-1)+(b_dnl[i-1]-Iu[0
187         }
188         b_dac_inl->SetBinContent(i, b_inl[i]-Iu[0]);
189         if (i>0) b_dac_dnl->SetBinContent(i-1, b_dnl[i-1]);
190     }
191     pad1->cd(12);
192     b_dac_inl->Draw();
193     pad1->cd(13);
194     b_dac_dnl->Draw();
195     if (update) c1->Update();
196     pad1->cd(17);
197     delta_dac->Draw();
198     pad1->cd(18);
199     b_cumulatif->Draw();
200     pad1->cd(11);
201     b_dac_ter->Draw();
202
203   } // end loop
204
205     for (int i=0 ; i<nop ; i++) { // calculate RMS
206         rms_dac->SetBinContent(i, sqrt(rms_dac->GetBinContent(i)/noofiteration4rms));
207         if (i<nop-1) rms_dnl->SetBinContent(i, sqrt(rms_dnl->GetBinContent(i)/noofiterat
208         rms_inl->SetBinContent(i, sqrt(rms_inl->GetBinContent(i)/noofiteration4rms));
209         b_rms_dac->SetBinContent(i, sqrt(b_rms_dac->GetBinContent(i)/noofiteration4rms))
210         if (i<nop-1) b_rms_dnl->SetBinContent(i, 1.55-3*(-Iu[0]/2+sqrt(b_rms_dnl->GetBin
211         b_rms_inl->SetBinContent(i, sqrt(b_rms_inl->GetBinContent(i)/noofiteration4rms))
212     }
213     pad1->cd(8);
214     rms_dac->Draw();
215     pad1->cd(9);
216     rms_inl->Draw();
217     pad1->cd(10);
218     rms_dnl->Draw();
219     pad1->cd(14);
220     b_rms_dac->Draw();
221     pad1->cd(15);
222     b_rms_inl->Draw();
223     pad1->cd(16);
224     b_rms_dnl->Draw();
225     if (update) c1->Update();
226
227   gBenchmark->Stop("binary_vs_thermometer");
228 }
```

Ln 216, Col 21 INS

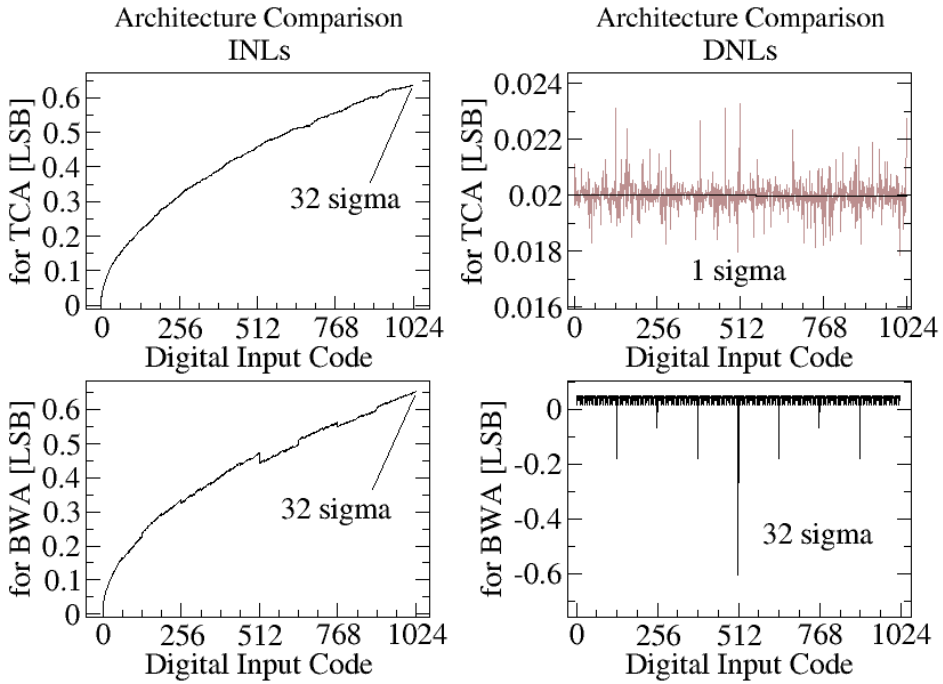
- *Betiğin işim taşımıyor olduğuna dikkat edin*
- *Nasıl başlayıp nasıl bittiğine dikkat edin*

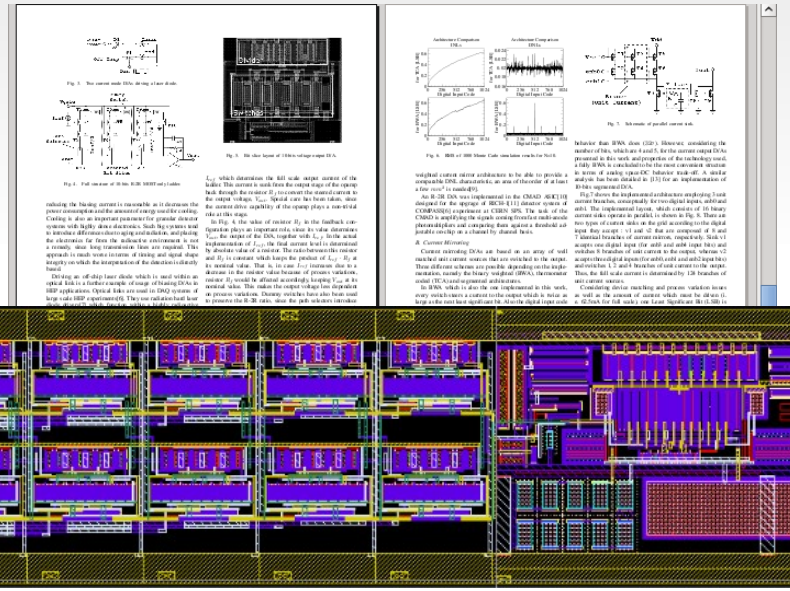
Kullanım

Betik yazmak

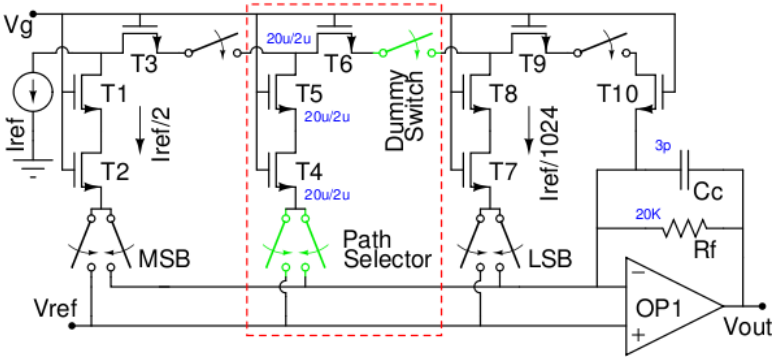


- *INL' ler çok farklı değil*
- *DNL' ler farklı, TCA' nin davranışı çok daha iyi, ama çok yer kaplayacak*
- *Seçimimi yaptım, D/A' ımı tasarladım ve gittiğim çalıştayda sundum*
- *Görev tamam !!*





Repeated
Bit Slice



Kullanım

Betiği *.so kitaplığına derlemek

- Mimari karşılaştırıcım betik halindeyken CINT tarafından *yorumlandığı* için yeterince hızlı sonuç vermeyebilir
- Onu derlediğimde ise *yorumlanmak* yerine, *çalışacaktır* ve bu çok daha hızlı bir süreçtir
- Betiğimde hemen hiç değişiklik yapmadan, onu bir *.so kitaplığına derleyebilirim (tıpkı *cizgi* ve *ok* class' larında olduğu gibi)
- Hemen hiç derken:
  - Kullanılan class' ların, başlık (header) kütüklerini eklemeliyim (*TCanvas* kullanılmış ise "`#include<TCanvas.h>`" gibi)
  - Yazdığım işleve (fonksiyon) tercihen betiğin saklandığı kütükle aynı olan bir isim vermeliyim (*isim.C* kütügü için "`int isim() {}`" gibi)

- Artık işlevimi *.so kitaplığına derlemeye hazırım:

```
> root mimariKarsilastirici2.C++
root [0] Processing mimariKarsilastirici2.C++...
Info in <TUnixSystem::ACLiC>: creating shared library
  /home/oc/Documents/HEP_Okulu/workDir/root/./mimariKarsilastirici2_C.so
```

- Kitaplığımı daha sonra tekrar kullanmak istediğimde:

```
oc@olmak2:~/Documents/HEP_Okulu/workDir/root$ root -l
root [0] .L mimariKarsilastirici2_C.so
root [1] mimariKarsilastirici2()
```

Kullanım

Betiği *.so kitaplığına derlemek

File Edit View Search Tools Documents Help

New Open Save Print... Undo Redo Cut Copy Paste Find Replace

mimariKarsilastirici.C | mimariKarsilastirici2.C

```
27 #include <TFile.h>
28 #include <TNtuple.h>
29 #include <TH2.h>
30 #include <TProfile.h>
31 #include <TCanvas.h>
32 #include <TFrame.h>
33 #include <TROOT.h>
34 #include <TSystem.h>
35 #include <TRandom.h>
36 #include <TBenchmark.h>
37 #include <TCint.h>
38 #include <TGraph.h>
39 #include <TF1.h>
40
41 int mimariKarsilastirici2() {
42    gROOT->Reset();
43
44    // edit these parameters for other simulations ----------
45    int noofbits=10;
46    int noofiteration4rms=100, nob=100; // nob : no of bins in distro
47    double sigma=0.02, centroid=1.0;
48    const Int_t kUPDATE = 50;
49    bool update=kTRUE;
50    //-------------------------------------------------------
51
52    int nop=pow(2.0, noofbits);
53    int boyut = pow(2, 3*noofbits);
54    double Iu[boyut]; // 2^noofbits current source
55    int lastincremented=0;
56    TCanvas *c1 = new TCanvas("c1","Binary vs Thermometer",1200,850);
57    TPad *pad1 = new TPad("pad1","Unit Current Source",0.0,0.0,1.0,1.0);
58    pad1->Draw();
59    pad1->Divide(5, 4, 0.001, 0.001);
60    pad1->cd(5)->SetFillColor(47);  pad1->cd(5)->SetGrid() ;
61    pad1->cd(6)->SetFillColor(47);  pad1->cd(6)->SetGrid() ;
62    pad1->cd(7)->SetFillColor(47);  pad1->cd(7)->SetGrid() ;
63    pad1->cd(8)->SetFillColor(47);  pad1->cd(8)->SetGrid() ;
64    pad1->cd(9)->SetFillColor(47);  pad1->cd(9)->SetGrid() ;
65    pad1->cd(10)->SetFillColor(47); pad1->cd(10)->SetGrid();
66    pad1->cd(11)->SetFillColor(21); pad1->cd(11)->SetGrid();
67    pad1->cd(12)->SetFillColor(21); pad1->cd(12)->SetGrid();
68    pad1->cd(13)->SetFillColor(21); pad1->cd(13)->SetGrid();
69    pad1->cd(14)->SetFillColor(21); pad1->cd(14)->SetGrid();
70    pad1->cd(15)->SetFillColor(21); pad1->cd(15)->SetGrid();
71    pad1->cd(16)->SetFillColor(21); pad1->cd(16)->SetGrid();
72
73    pad1->cd(1);
```

Ln 1, Col 1 INS

File Edit View Search Tools Documents Help

New Open Save Print... Undo Redo Cut Copy Paste Find Replace

mimariKarsilastirici.C | mimariKarsilastirici2.C

```
198        if (i>0 && i<nop) {
199            b_dnl[i-1]=Iu[0]-b_dac[i]+b_dac[i-1];
200            b_rms_dnl->SetBinContent(i-1, b_rms_dnl->GetBinContent(i-1)+(b_dnl[i-1]-Iu[0
201        }
202        b_dac_inl->SetBinContent(i, b_inl[i]-Iu[0]);
203        if (i>0) b_dac_dnl->SetBinContent(i-1, b_dnl[i-1]);
204    }
205    pad1->cd(12);
206    b_dac_inl->Draw();
207    pad1->cd(13);
208    b_dac_dnl->Draw();
209    if (update) c1->Update();
210    pad1->cd(17);
211    delta_dac->Draw();
212    pad1->cd(18);
213    b_cumulatif->Draw();
214    pad1->cd(11);
215    b_dac_ter->Draw();
216
217  } // end loop
218
219   for (int i=0 ; i<nop ; i++) { // calculate RMS
220       rms_dac->SetBinContent(i, sqrt(rms_dac->GetBinContent(i)/noofiteration4rms));
221       if (i<nop-1) rms_dnl->SetBinContent(i, sqrt(rms_dnl->GetBinContent(i)/noofiterat
222       rms_inl->SetBinContent(i, sqrt(rms_inl->GetBinContent(i)/noofiteration4rms));
223       b_rms_dac->SetBinContent(i, sqrt(b_rms_dac->GetBinContent(i)/noofiteration4rms))
224       if (i<nop-1) b_rms_dnl->SetBinContent(i, 1.55-3*(-Iu[0]/2+sqrt(b_rms_dnl->GetBin
225       b_rms_inl->SetBinContent(i, sqrt(b_rms_inl->GetBinContent(i)/noofiteration4rms))
226    }
227    pad1->cd(8);
228    rms_dac->Draw();
229    pad1->cd(9);
230    rms_inl->Draw();
231    pad1->cd(10);
232    rms_dnl->Draw();
233    pad1->cd(14);
234    b_rms_dac->Draw();
235    pad1->cd(15);
236    b_rms_inl->Draw();
237    pad1->cd(16);
238    b_rms_dnl->Draw();
239    if (update) c1->Update();
240
241  gBenchmark->Stop("binary_vs_thermometer");
242  return 0;
243 }
244
```

Ln 1, Col 1 INS

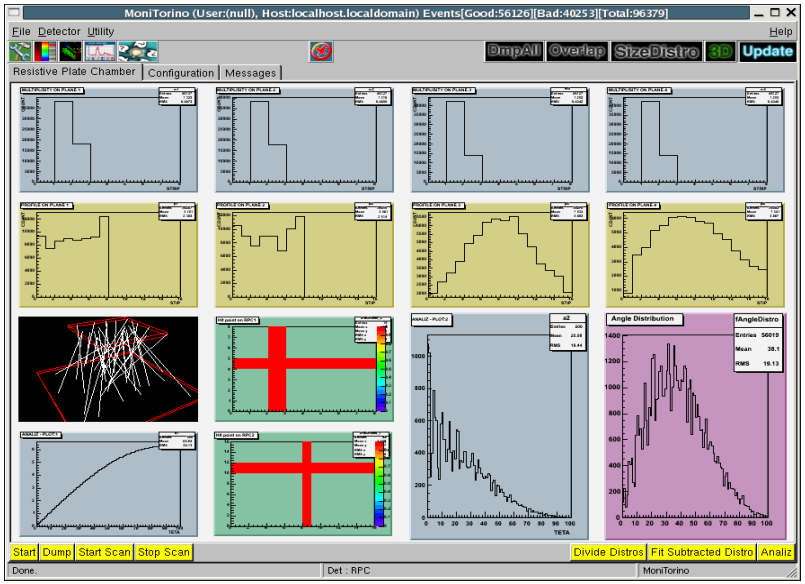
- *Betiğin isim taşıdığına ve kullanılan kitaplığın başlık kütüklerinin içerildiğine dikkat edin*
- *Bir tamsayı döndürülmeli*

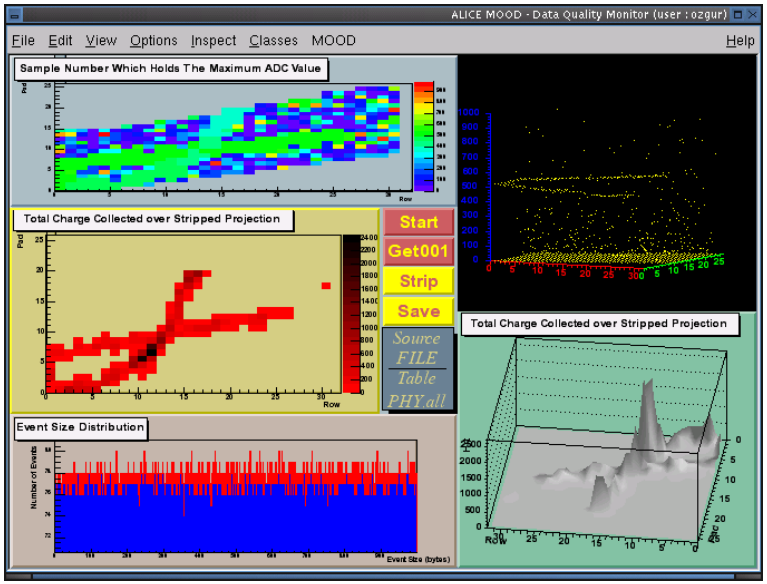
Kullanım

Bağımsız derlemek (uygulama yazmak)

- Mimari karşılaştırıcım *.so haline geldiğinde, ROOT ortamında ".L" ile yüklenebilecek ve sanki bir ROOT komutuymuş gibi çalişitirilabilecektir
- Daha yüksek verimle çalısacaktır
- ROOT' un varlığına ihtiyaç duyacak ve ancak uygun bir ROOT kitaplığı ile oluşturulabilecek/çalıştırılabilecektir.

- Halbu ki ROOT ortamına ihtiyaç duymaksızın, ROOT' u bir kitaplık olarak kullanmak (tıpkı *cizgi* ve *ok* kitaplığını kullandığımız gibi) ve kendi ayakları üzerinde duran (standalone) bir uygulama (application) geliştirmek de mümkün.

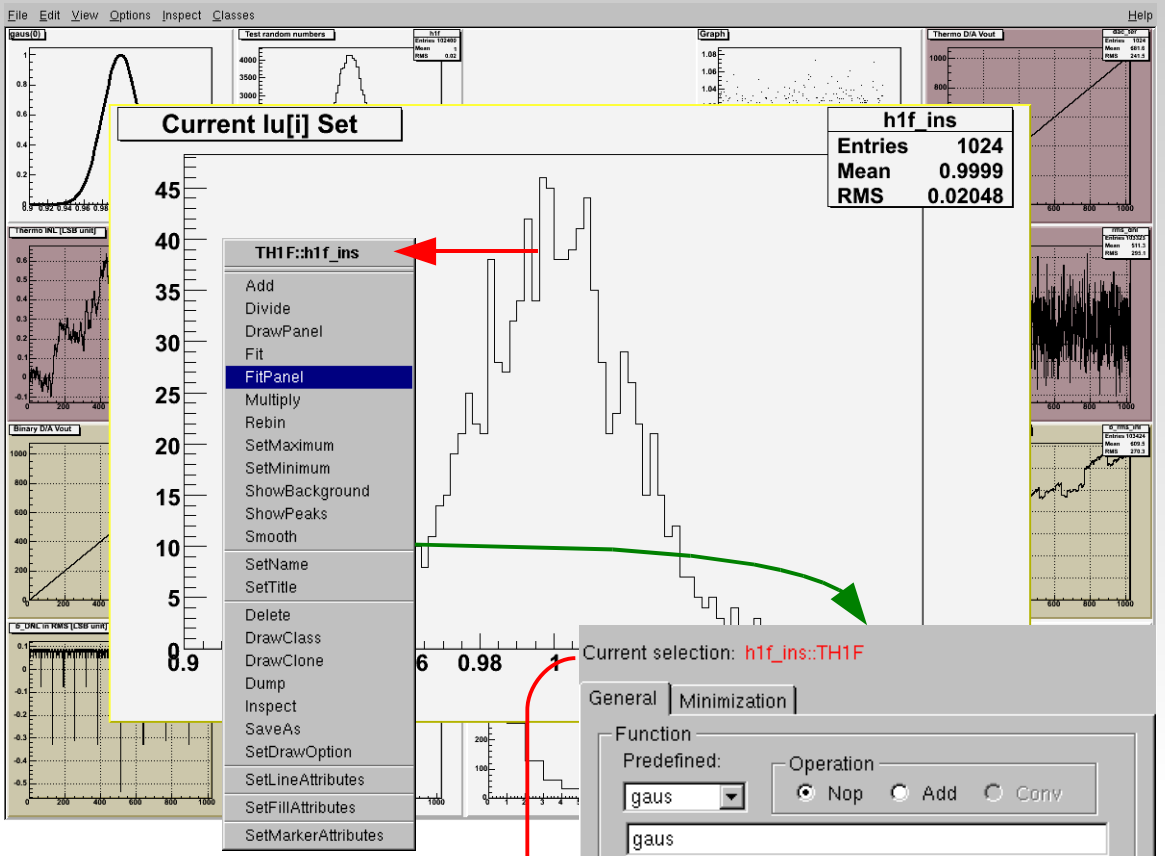


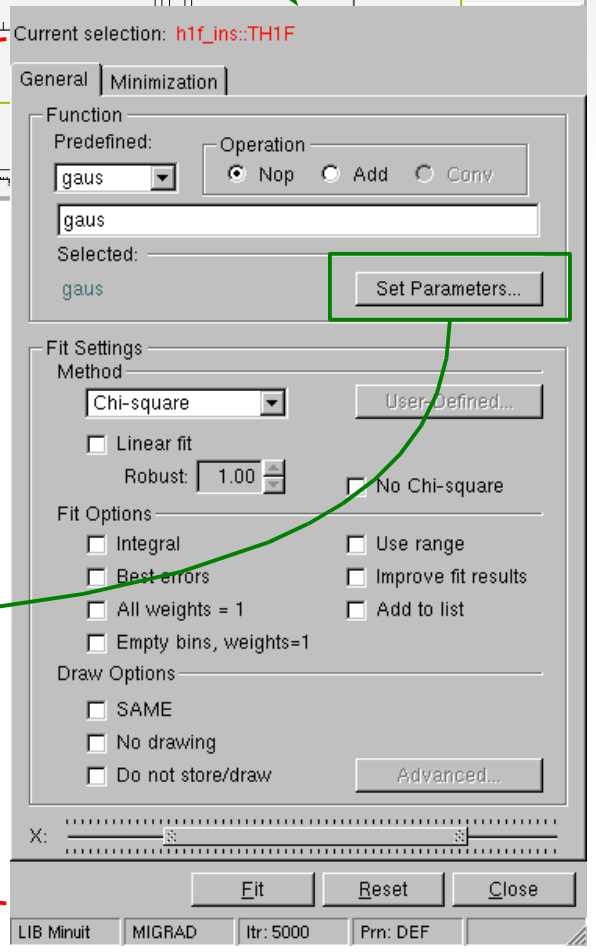


Kullanım

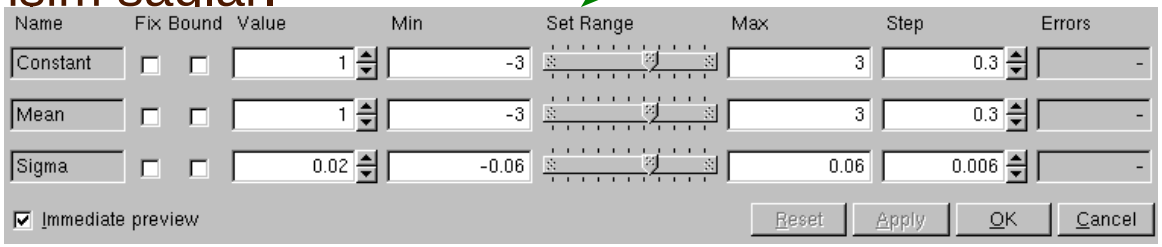
GUI Kullanmak

- Açılan tüm pencereler ve bu pencerelerde görünen ***HERŞEY ama HERŞEY*** ya ROOT class kitaplığından yaratılmış NESNELER' dir (histogramlar, eksenler, başlık, alt-pencereler v.b.) ya da ROOT class kitaplığı aracılığı ile ulaşılabilecek başka kitaplıklardan yaratılmış başka nesnelerdir (x11 kitaplığından yaratılan nesneler ve pencere yönetici işlevleri gibi).
- Sağ tıklama, öbekte (heap) oluşturulmuş bu nesnelerin üye işlevlerinden (class member function) bazılarına erisim sağlar.



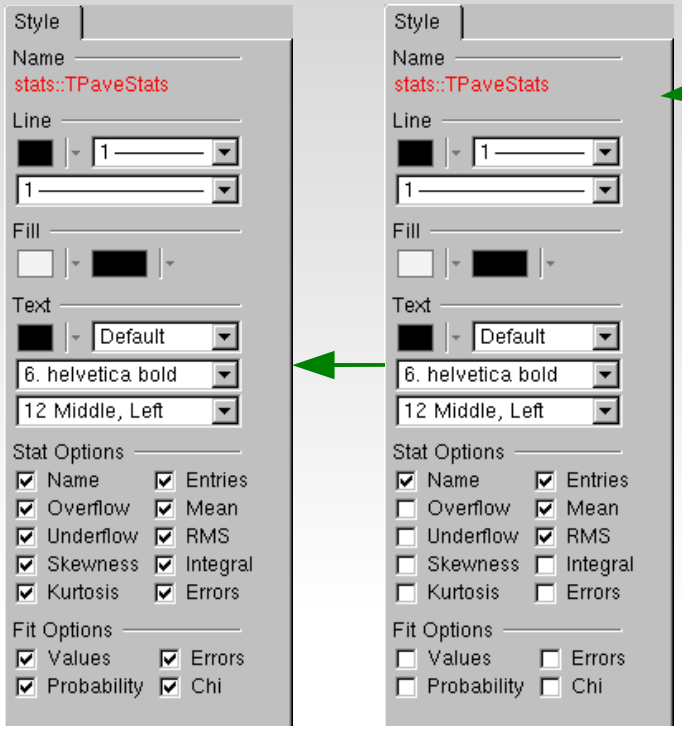


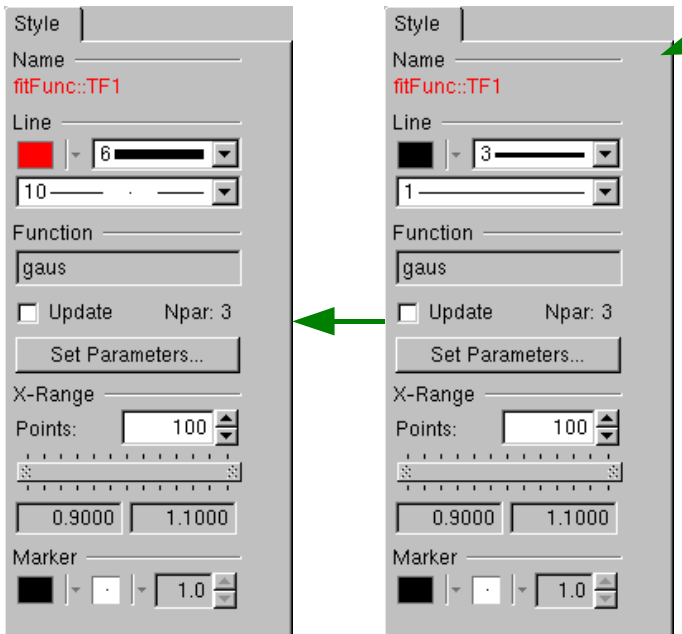
- *Ön tanımlı uyum işlevleri (fit function)*
- *Pekçok değişken*
- *Hesap aralığı*

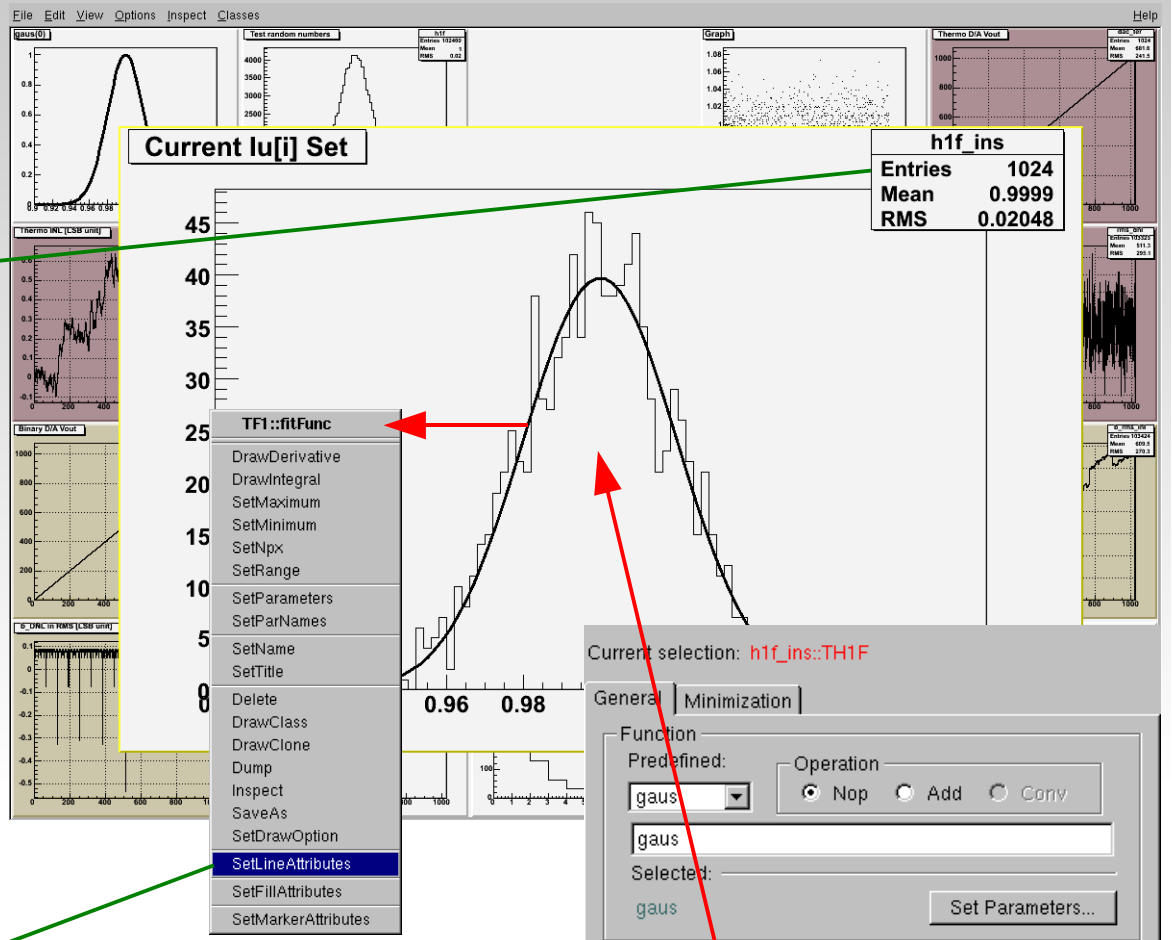


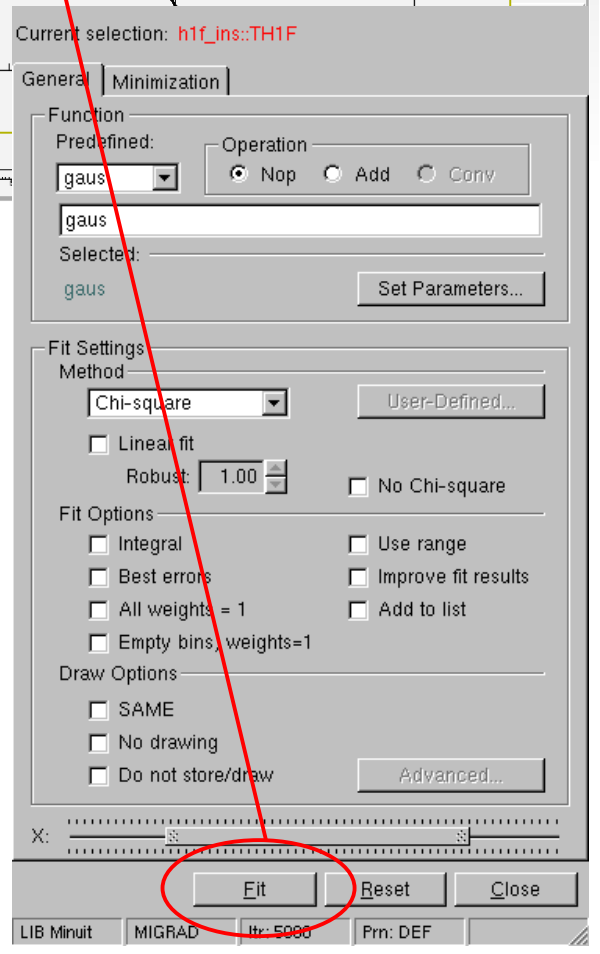
Kullanım

GUI Kullanmak









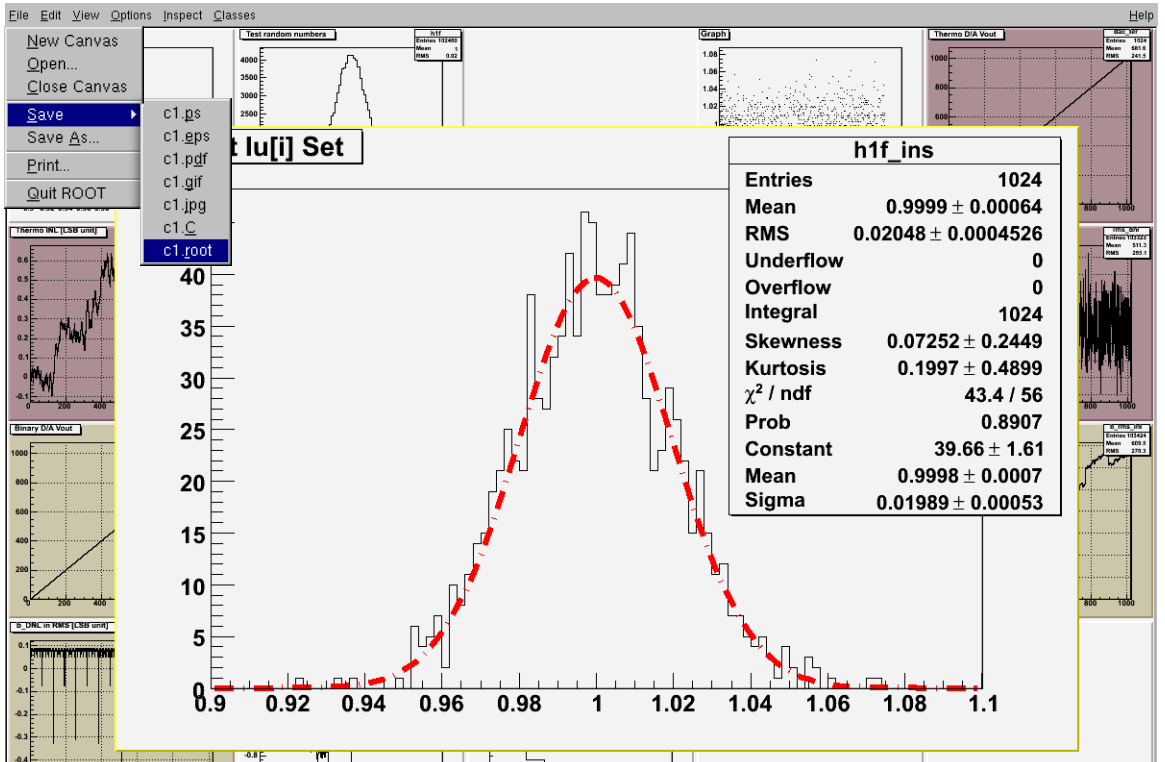
- *Uyum eğrisi çizildikten sonra, süslemek ve/veya veriyi daha görünür kılmak mümkün*
- *Uyum eğrisinin daha **kalın**, **kesikli** ve **kırmızı** olmasi ve*
- *Histogramla ilgili daha fazla istatistik bilginin görünür kılınması gibi*

Kullanım

GUI Kullanmak

- Veri çözümlemesi (analiz) yapan yazılımımız (betik) başarı ile çalıştı, sonuç üretti, kullanıcı sonuç grafikleri üzerinde gerekli düzeltmeleri yaptı veriyi daha okunur hale getirdi.
- Yapılan ***çalış***mayı ***sakla***maya geldi sıra = ***kaydet***mek !
- Farklı yollar mevcut; eldeki nesne:
  - Bir *.C kaynak kütüğü olarak saklanabilir
    - Bu kütüğü açmak, root ile tekrar yorumlamayı gerektirir (ör. "`root -l k.C`")
  - Bir *.root kütüğü olarak saklanabilir
    - Bu kütüğün içeriğine ulaşmak, bir TBrowser nesnesi ile doğrudan mümkündür (ör. "`TBrowser a`")
  - Resim olarak da saklanabilir: ps, eps, gif v.b.

- *Uyum eğrisi daha **kalın**, **kesikli** ve **kırmızı***
- *Histogramla ilgili daha fazla istatistik bilgi görülüyor*

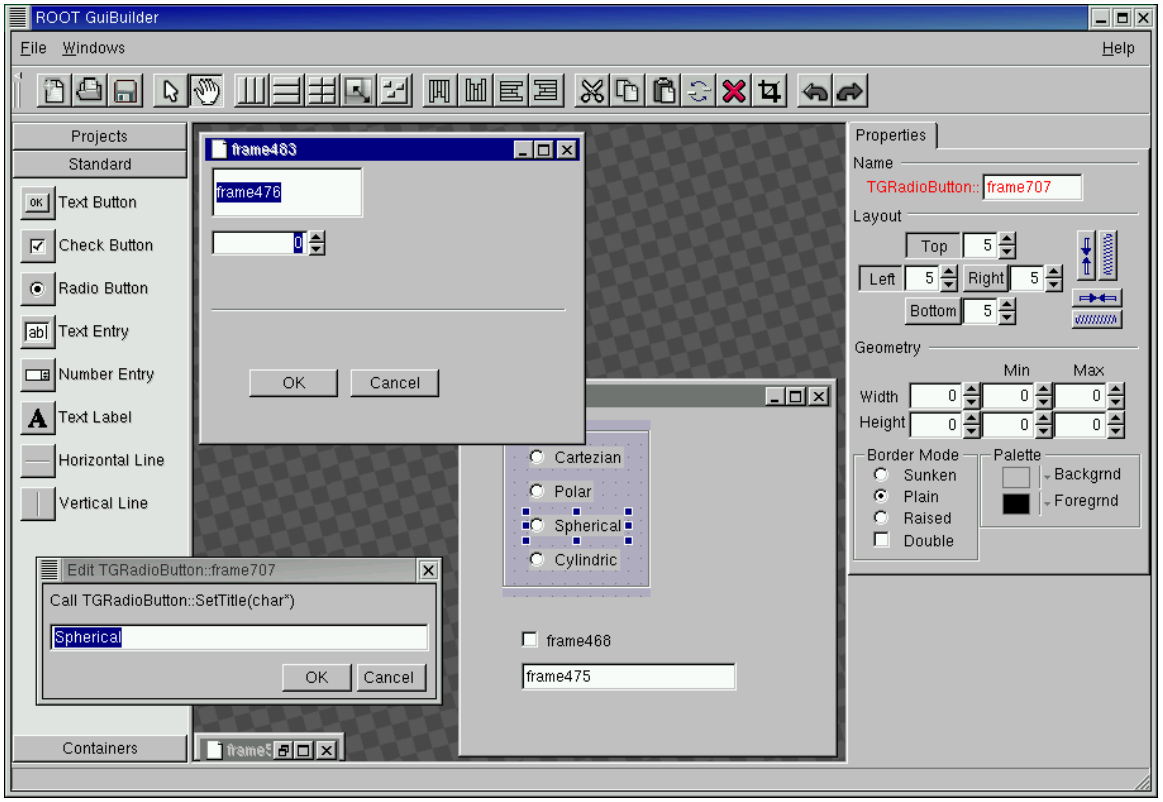


Örnek

GUI Yaratmak

- Kullanıcılar ROOT GUI class kitaplığını kullanarak, yazdıkları uygulamalara grafik kullanıcı arayüzü (GUI veya HI) yazabilirler.
- Bunu yapmanın temelde iki yolu vardır ve çoğunlukla dönüşümlü olarak beraber kullanılırlar:
  - Kaynak yazarak
  - TRootGuiBuilder class' ının bir nesnesini kullanarak

```
oc@olmak2:~/Documents/HEP_Okulu/workDir/root$ root -l
root [0] TRootGuiBuilder a
root [1] _
```



- Görsel olarak hazırlanan kullanıcı arayüzü, ROOT betiği olarak, *.C uzantısı ile kaydedilir ve asıl işi yapacak kaynağın geliştirilmesine bu kütük üzerinde devam edilir.
- ***Çok eğitici: yap, kaydet, oku !!***
- Zaman kazandırıcı: ROOT GUI class' larını ezbere bilme gerekliliği yok
- ***Sık sık kaydedin, arada göçebilir !!***

ROOT' ta Hayatta Kalmak

Kullanım kılavuzunun kullanımı

- Yeni bir kitaplık (library) ile tanışmak dört ayak üzerine oturur : *(sanırım)*
  i. Kullanım kılavuzu (user manual)
  ii. Kaynak belgelendirmesi (source code documentation)
  iii. Kitaplığın kullanımını gösteren örnekler topluluğu (tutorial, test, example)
  iv. Sizin o örnekleri çalığıp kendi uygulamalarınızı yazmaya başlamanız

- ***Şanslıyız:* ROOT bunların hepsine sahip !! ROOT' la tanışmak zor değil !!**

- ***Kullanım kılavuzu***, kitaplığın hangi düşüncelerle nasıl oluşturulduğunu, kurulum ve “merhaba dünya” başlangıcının nasıl yapılacağını gösterir.
- Hızlı değişmez, durağandır.
- En başta sadece bir kez düzgün ve sindirerek çalışılmalı.
- Pek çok geliştirici için her yeni sürümde, yüzeysel göz gezdirmek yeterli

- Bu derste anlatılanlardan çok daha ayrıntılı bilgiye, akıcı bir dil ile yazılmış kullanım klavuzundan ulaşmak mümkün

ROOT' ta Hayatta Kalmak

Kullanım kılavuzunun kullanımı

- Yeni bir kitaplık (library) ile tanışmak dört ayak üzerine oturur : *(sanırım)*
  i. Kullanım kılavuzu (user manual)
  ii. Kaynak belgelendirmesi (source code documentation)
  iii. Kitaplığın kullanımını gösteren örnekler topluluğu (tutorial, test, example)
  iv. Sizin o örnekleri çalışıp kendi uygulamalarınızı yazmaya başlamanız

- ***Şanslıyız:* ROOT bunların hepsine sahip !! ROOT' la tanışmak zor değil !!**

- ***Kullanım kılavuzu***, kitaplığın hangi düşüncelerle nasıl oluşturulduğunu, kurulum ve "merhaba dünya" başlangıcının nasıl yapılacağını gösterir.
- Hızlı değişmez, durağandır.
- En başta sadece bir kez düzgün ve sindirerek çalışılmalı.
- Pek çok geliştirici için her yeni sürümde, yüzeysel göz gezdirmek yeterli

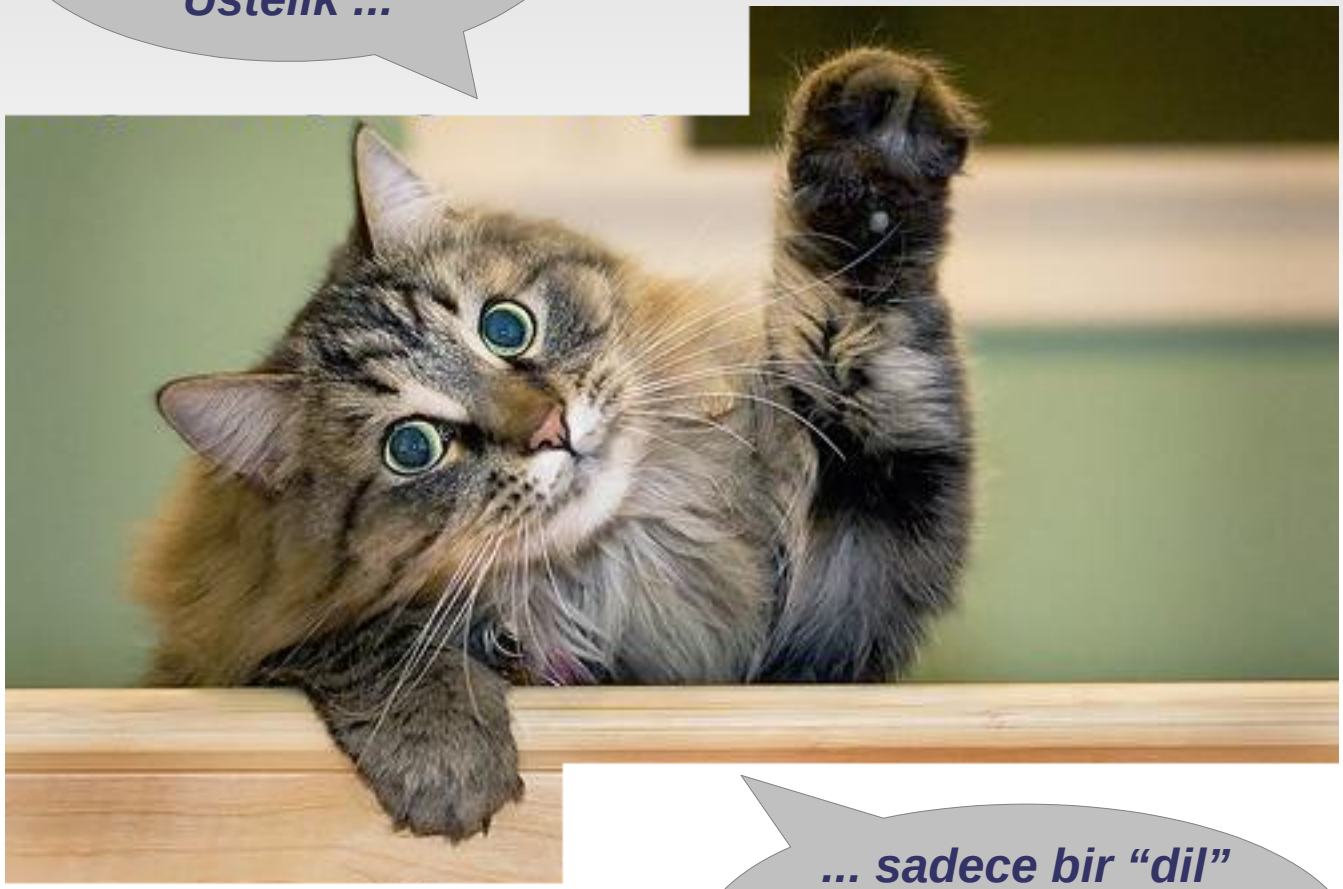
- Bu derste anlatılanlardan çok daha ayrıntılı bilgiye, akıcı bir dil ile yazılmış kullanım klavuzundan ulaşmak mümkün

* Lolcat' ten

ROOT' ta Hayatta Kalmak

Kullanım kılavuzunun kullanımı

ROOT kullanım kılavuzu ~500 sayfadır !
Üstelik ...



... sadece bir “dil” öşretir, o dilde ne söyleneceğini değil !

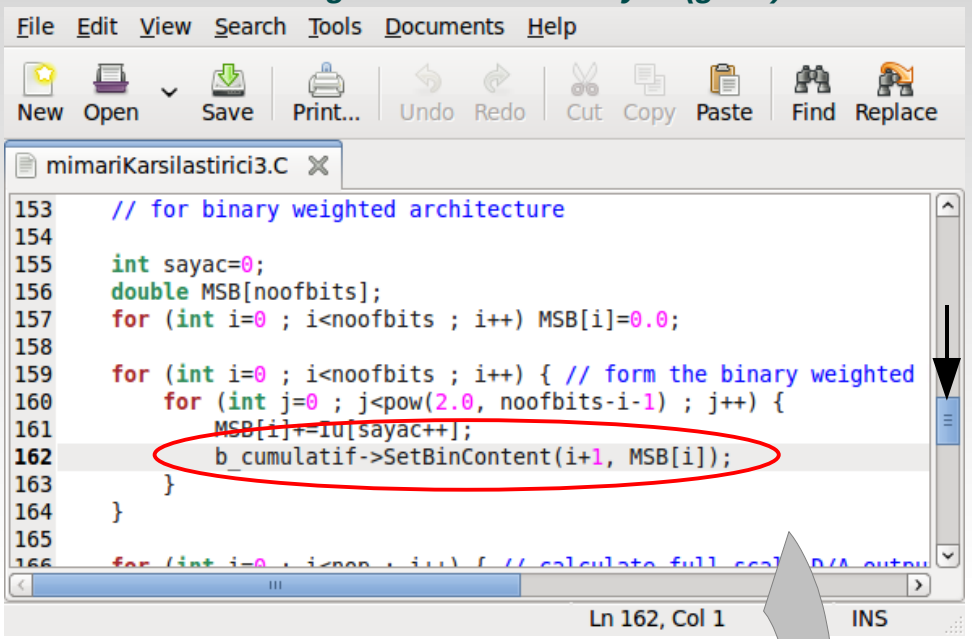
* Lolcat' ten

ROOT' ta Hayatta Kalmak

HTML belgelendirmenin kullanımı

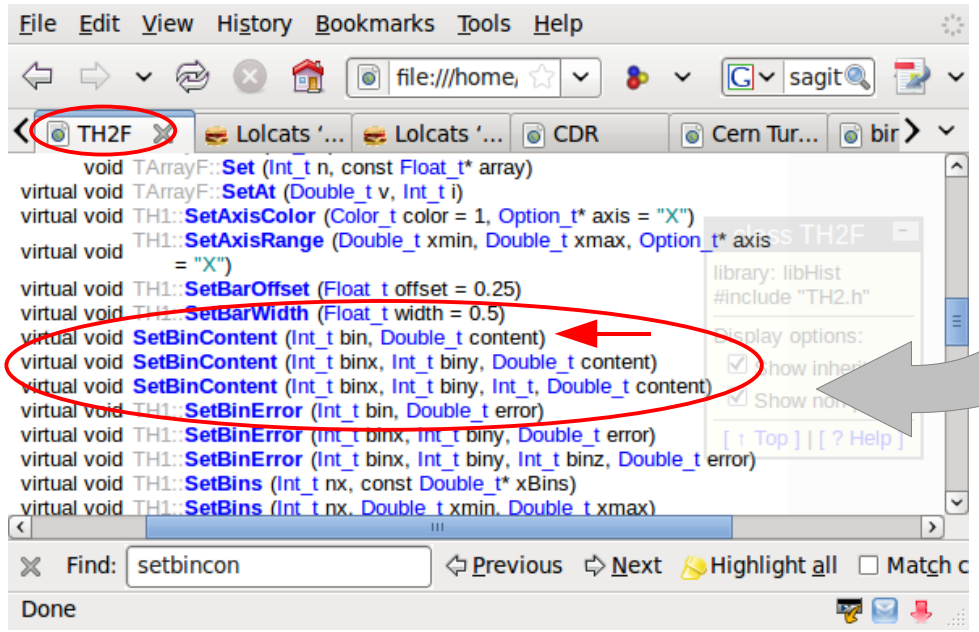
- ***Ölümle kalım*** arasındaki çizgi
- Öğrenilecek/çalışılacak bir şey ***değil***, ROOT' arken devamlı olarak başvurulacak en düşük seviyeli (asıl olana en yakın mesafede) belgelendirme
- ***Canlı:*** kullanılıyor olan ***ROOT sürümünden doğrudan üretiliyor***, dolayısıyla:
  - ***Kullanım klavuzları*** gibi ***durağan*** (statik), dolayısıyla ***eskimiş bilgilerle dolu*** ve bu anlamda ***kullanışsız*** değil
  - Kullanılan kitaplığın ***kaynağını okumak***la eşdeğer, ***hataya en az açık*** yöntem

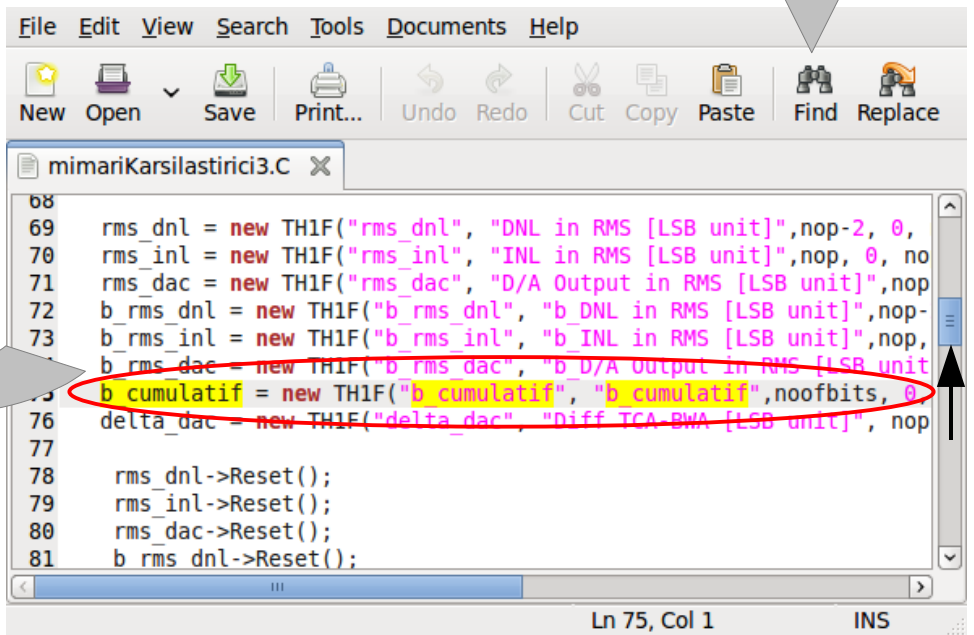
En sevdiğiniz metin düzenleyici (gedit)



Ctrl-F ile "b_cumulatif" aranıyor ve "ne" olduğuna bakılıyor: TH1F nesnesi

En sevdiğiniz html yorumcu (firefox)





ROOT' ta Hayatta Kalmak

$ROOTSYS/tutorials dizininin etkin işlevi

- ***Betik*** yazarken ***ölümle kalım*** arasındaki çizgi
- Öğrenilecek/çalışılacak bir şey ***değil***, ROOT' arken devamlı olarak başvurulacak en düşük seviyeli (asıl olana en yakın mesafede) öğretmen

- ***Yeni bir kitaplık kullanmak***, özellikle C/C++ gibi kitaplık desteği olmadan herşeyin zor olduğu dillerde, ***yeni bir dil öğrenmek***tir.
- Dil ***konuş***ularak öğrenilir.
- Bu dizin, birlikte ROOT' ça ***konuşmaya başlanacak arkadaşları***mızın yaşadığı yer

- ROOT ile yeni tanışmada bu dizindeki ***TÜM ama TÜM alıştırmaları***, açın, ne yapmak istediğini yüzeysel olarak anlayın ve çalıştırın.
- Çünkü:
  - Alıştırmaların ne kadarının çalıştığını ne kadarının çalışmadığını kendi tecrübenize dayanarak öğrenmelisiniz – kullanacağınız ***kitaplık mükammel mi*** ?
  - Bu alıştırmalar, ileride yazacağınız uygulamalar için ***başlangıç örnekleri*** içeriyor.
  - Hangi *.C kütüğünün hangi işlevi yerine getirdiğini hatırlamanız gerek***mi***yor.
  - Sadece ***“tutorial' da bu işi yapan bir alıştırma vardı, oradan başlayabilirim...”*** diyebilmeniz gerekli

ROOT' ta Hayatta Kalmak

$ROOTSYS/test dizininin etkin işlevi

- ***Uygulama*** yazarken ***ölümle kalım*** arasındaki çizgi
- Öğrenilecek/çalışılacak bir şey ***değil***, ROOT' arken devamlı olarak başvurulacak en düşük seviyeli (asıl olana en yakın mesafede) öğretmen

- ***Yeni bir kitaplık kullanmak***, özellikle C/C++ gibi kitaplık desteği olmadan herşeyin zor olduğu dillerde, ***yeni bir dil öğrenmek***tir.
- Dil ***konuş***ularak öğrenilir.
- Bu dizin, birlikte ROOT' ça ***konuşmaya başlanacak arkadaşları***mızın yaşadığı yer

- ROOT ile yeni tanışmada bu dizindeki ***TÜM ama TÜM uygulamaları***, açın, ne yapmak istediğini yüzeysel olarak anlayın, derleyin ve çalıştırın.
- Çünkü:
  - Uygulamaların ne kadarının çalıştığını ne kadarının çalışmadığını kendi tecrübenize dayanarak öğrenmelisiniz – kullanacağınız ***kitaplık mükammel mi*** ?
  - Bu uygulamalar, ileride yazacağınız uygulamalar için ***başlangıç örnekleri*** oluşturuyor.
  - Hangi *.cxx (ve karşılık gelen *.h) kütüğünün hangi işlevi yerine getirdiğini hatırlamanız gerek***mi***yor.
  - Sadece ***"test' te bu işi yapan bir alıştırma vardı, oradan başlayabilirim..."*** diyebilmeniz gerekli

Örnek

İlk komutun uygulama hali

ilkUygulama.cxx | TTPLOTist.cxx | TTPLOTist.h

```
1 #include "TApplication.h"
2 #include "TCanvas.h"
3 #include "TLine.h"
4 #include "TF3.h"
5
6 int main(int argc, char **argv)
7 {
8     // ROOT uygulama nesnesi yaratiliyor
9    TApplication benimGuzelUygulamam("App", &argc, argv);
10
11    // Bir pencere (veya kanvas) olusturup, isletim sisteminin pencere
12    // yönecisi arasinda signal-slot haberlesmesini baslat
13   TCanvas *kanvas = new TCanvas("kanvas", "Benim Guzel Kanvasim", 500, 400);
14   kanvas->Connect("Closed()", "TApplication", &benimGuzelUygulamam, "Terminate()");
15
16    // Komut örnegimizde yaptigimizin aynisini yapiyoruz.
17   TF3 *islev = new TF3("islev","sin(cos(x)+sin(y)+cos(z))",-10,10,-10,10,-10,10);
18
19    // Nesnenin, kendisini görünür kilan üye islevini çagir
20   islev->Draw();
21
22    // Kanvas nesnesinin, üzerindeki nesnelerin görünürlük
23    // durumunu yenileyen üye islevini çagir
24   kanvas->Update();
25
26    // Uygulamayi kendi halinde çalismaya birak (GUI event loop' a sok)
27   benimGuzelUygulamam.Run();
28
29   // Sonlandirildiginda isletin sistemine basari isareti göndererek çik
30   return 0;
31 }
```

Kullanılan kitaplıkların başlıkları eklenmiş

Alışık olduğumuz C/C++ başlangıcı

Ln 13, Col 69 INS

Örnek

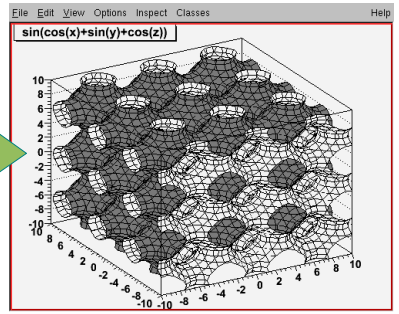
İlk komutun uygulama hali

- *Uygulama derlemek için en seçkin yol "Makefile" yazmaktır fakat burada bunu yapmayacağız.*

- Derlemek için:
  - **g++ -L/usr/lib/root -lCore -lCint -lRIO -lNet -lHist -lGraf -lGraf3d -lGpad -lTree -lRint -lPostscript -lMatrix -lPhysics -lz -pthread -lm -ldl -rdynamic -pthread -m64 -I/usr/include/root -L/usr/lib/root -lCore -lCint -lRIO -lNet -lHist -lGraf -lGraf3d -lGpad -lTree -lRint -lPostscript -lMatrix -lPhysics -lz -lGui -pthread -lm -ldl -rdynamic ilkUygulama.cxx -o ilkUygulama**

- *Bu kadar seyi hatırlamak zor:*
  - ***root-config**: ROOT kütüphanesi kullanıcıları için hayatı kolaylaştırmak için yazılmış bir komut satırı uygulamasıdır (her kitaplığın böyle kullanışlı bir yardımcı uygulaması olur)*
  - *Uygulamanın derlenmek için ihtiyaç duydugu gerekli kitaplıkları döndürür*
  - *Çoğunlukla " ` " tırnakları arasında kullanılır (Linux ve Uygulamalar – I dersinden hatırlayın)*
  - **g++ `root-config --glibs --cflags` ilkUygulama.cxx -o ilkUygulama**

```
oc@olmak2:~/HEP_Okulu/root$ ./ilkUygulama &
[1] 16811
oc@olmak2:~/HEP_Okulu/root$ _
```



ROOT - I

Onunla hayatta kalma != Onda hayatta kalma != Ona rağmen hayatta kalma

- Giriş
  - Nedir ?
  - Neden iyidir ?
- Kullanım
  - Komut vermek
  - Betik yazmak
  - Betiği *.so kitaplığına derlemek
  - Bağımsız derlemek (uygulama yazmak)
  - ROOT GUI' sını kullanmak
  - Kullanıcı uygulamaları için GUI yaratmak
- ROOT' ta Hayatta Kalmak
  - ROOT kullanım klavuzu' nun kullanımı
  - HTML kaynak belgelendirmesinin kullanımı
  - $ROOTSYS/tutorials dizininin etkin işlevi
  - $ROOTSYS/test dizininin etkin işlevi
- Örnek
  - “Kullanım - Komut vermek” kısmındaki örneğin uygulama hali
- Akşam sefası
  - Ertesi günün sabahına hazırlanması beklenen akşamlık ödevler

Aksam Sefası

Ertesi günün sabahına hazırlanması beklenen akşamlık ödevler

- *TH1F olarak tanımlandığını gördüğünüz BenimGuzelHistogramim isimli bir nesne:*
  - *"`BenimGuzelHistogramim->ClassName()`" biçiminde kullanılmış, bu kaynak parçasının ne iş yaptığını açıklayın.*