

1 Izvodjenje naredbi u programu *Root*

Nakon pokretanja programa *Root*, naredbe se mogu izvoditi interaktivno ili upisane u skriptu (iz programa se izlazi naredbom `.q`).

U slučaju da su naredbe upisane u skriptu izvode se naredbom `.x ime.C` (gdje je *ime* naziv skripte, s nastavkom `C`), s tim da skripta treba imati određenu formu i početi, u slučaju da je naziv skripte `mion.C`, s:

```
void mion(){
Root naredbe
// komentari
...
}
```

Root naredbe se nalaze između zagrada, tekst upisan iza znakova `//` *Root* ne uzima u obzir, komentari. Ako se upisuju u skriptu naredbe moraju završavati s `;`.

2 Prikazivanje i operacije s podacima u histogramima

2.1 Površina za prikazivanje (Canvas)

- Definicija nove površine za prikazivanje (Canvas):

```
TCanvas *time = new TCanvas("c1","mion",100,130,1000,600);
```

- Neke naredbe vezane za površinu za prikazivanje:

```
// podjela površine za prikazivanje na dva dijela
time->Divide(2,1);
// prijelaz u prvi dio površine za prikazivanje
time->cd(1);
```

2.2 Histogrami

- Definicija histograma:

```
TH1F *h1 = new TH1F("h1","tdc_kanal",4096,0,4096);
\\ h1 je histogram s 4096 podjeljaka (binova), početni
\\podjeljak je 0 a maksimalni 4096.
```

- Crtanje histograma:

```
// Crtanje histograma s imenom h1
h1->Draw();
```

- Primjer učitavanja podataka iz file mion.dat u definirani histogram s imenom h1:

```

    ifstream in;
    Float_t t;
    Int_t nlines= 0;

    in.open("mion.dat", ios::in);

    while (1) {
        in >> t;
        if (!in.good()) break;
        h1->Fill(t);
        nlines++;
    }

    in.close();

```

2.3 Operacije s podacima u histogramima

- Primjer grupiranja podataka (rebin):

```
TH1F *h2 = h1->Rebin(40, "h2");
```

- Primjer prilagodbe funkcije na podatke (fit):

```

// definicija eksponencijalne funkcije:
TF1 fit1("fit1", "[0]*TMath::Exp(-x/[1])");
//
fit1.SetParameter(1,2);
h2 -> Fit("fit1", "RLE", "", 100, 3500);
time->cd(2);
h2->Draw();

```

- Primjer promjene horizontalne skale (potrebno napr. za prijelaz prikaza kanal → vrijeme):

```

// odredjivanje broja biniranja:
int nbins = h2->GetXaxis()->GetNbins();
// definicija histograma s novom skalom x-osi
TH1F *h3= new TH1F("h3", "vrijeme", nbins, 0, 5);
// odredjivanje broja dogadjaja za svaki bin:
for (int i=1; i<=nbins; i++) {
// odredjivanje broja dogadjaja u svakom binu:
    double y = h2->GetBinContent(i);
// odredjivanje sredine svakog bina:
    double x = h2->GetXaxis()->GetBinCenter(i);
// transformacija skale x-osi:

```

```
// kanal -> mikrosekunda (1ch = 1.25ns)
    double xnew = 0.00125*x;
// histogram s novom skalom x-osi:
    h3->Fill(xnew,y);
}
```