4-й семестр, lesson3. ROOT and 1-dimensional histograms

- **Several useful commands with vi editor**
- **Movement in buffer, “ :10 “ means move to 10th line**
- **Go to Insert mode : command I (insert) or R (replace)**
- **Return from Insert mode – command “ ^[“**
- **Save result end exit – edit from input mode, then command ZZ**
- **Exit without save – command :q!**
- **Movement in buffer: use arrows or letters: j – down , k – up, h – left, l - right**
- **Delete 3 lines – 3dd; deleted lines are taken into buffer which can be used with commands : P – insert above current line; or p – insert after current ine**
- **Open other file – command “ :e! file2”, if a buffer was filled in work with file1, it can be used now in work with file2**
- **Replace ”text1” to “other” in all lines -> :1,$s?text1?other? Carriage return**

Input/output and weighted histograms

- open **input** file:
- TFile *_file0 = TFile::Open("input_file.root" ;

- some details:
- 1)
- if **weighted** histogram needed,
- then the error on bin content is calculated
- as sqrt (sum of weight*weight);
- a supplementary statement needed just after histogram definition,
- h1 -> Sumw2();
- and the weight value should be supplied for event:
- **h1 -> Fill(value, weight);**

Canvas and Pads

- 2) The basic whiteboard in which an object is drawn in ROOT is called a **canvas** (defined by the class **TCanvas**). Every object in the canvas is a graphical object in the sense that you can grab it, resize it, and change some characteristics using the mouse.
- The canvas area can be divided in several sub areas, so-called **pads** (the class **TPad**). A pad is a canvas sub area that can contain other pads or graphic objects.
- At any one time, just one pad is the so-called active pad. Any object at the moment of drawing will be taken in the active pad.
- The obvious question is what is the relation between a canvas and a pad? In fact, a canvas is a pad that spans throughout the entire window. This is nothing else than the notion of inheritance. The **TPad** class is the parent of the **TCanvas** class.
- In ROOT, most objects derive from a base class **TObject**. This class has a virtual method Draw() such as all objects supposed to be "drawn". If several canvases are defined, there is only one active at a time.
- One draws an object in the active canvas by using the statement: **object.Draw**(). This instructs the object "object" to be drawn.

Objects and directories

- If no canvas is opened, a default one (named "**c1**" is created.
- The following command does a subdivision of canvas to pads:
- **c1 -> Divide(3, 2**). It creates 6 pads (3 on X by 2 on Y dimension).
- Canvas can be cleaned from previous objects: **c1 -> Clean().**
- Activation of a pad is performed by command **c1->cd(1),** ..., c1->cd(6).
- After filling, a canvas can be written to file in different formats:
- **c1 -> SaveAs**("file_name", "pdf")
- c1 -> SaveAs("file_name", "png")
- c1 -> SaveAs("file_name", "eps").
- 3) Histograms and other objects can be arranged in directories,
- like usual files. The " **.ls**" command shows content of current directory, "cd" command provides change the current directory.

Various comments

- 4) Objects can be written to **file** or kept in **memory**, and in some cases this difference does matter.
- The ".ls " command **.ls** marks as "**KEY**" the object in file and the objects in memory as "**OBJ**".
- 5) There are **tips** in histograms, canvases and other classes, which indicate possible actions with object, in the same spirit as LINUX provides tips for file names after indication of first characters and then pushing the TAB.
- 6) **Warning:** if the commands are written in file and it has structure like
- #include "TH1"
- void example(){
- ...
- }
- then the **name of the file** should be the same as the **name of the main** segment, i.e. both should be called " example ".

Exercise 2

- **Log-in** to terminal (for example to lfilk7@lk7-408
- **cd** student/20160215/
- there is **a link** "ntbeam_cher_r17_1_v15.root" to NTuple with data file and an example of a program in file "**example_2_edited.C**"
- It **reads** the input Ntuple, gets the number of reconstructed π^0 in event and **fills** a corresponding histogram.
- Try to start ROOT and run the ROOT command file:
- **Root**
- **.X example_2_edited.C**
- .q
- The identifier of the histogram is h_001 . Before leaving the ROOT, one can do several exercsises:
- cout << **h_001->GetEntries()** << endl;
- cout << **h_001->GetBinContent(1)** << endl;

Exercise 2

- cout << **h_001->GetBinContent(2)** << endl;
- c1->**SetLogy();**
- c1->SetLogy(0);
- cout << h_001->GetBinError(2) << endl;
- h_001->GetXaxis()->SetTitle("Mass, GeV");
- h_001->SetLineColor(kRed);
- h_001->Draw();
- Open Output file and write the h_001 into output file.
- More details can be found in https://root.cern.ch/guides/users-guide
- https://root.cern.ch/root/htmldoc/guides/users-guide/Introduction.html
- https://root.cern.ch/root/htmldoc/guides/users-guide/GettingStarted.html
- https://root.cern.ch/root/htmldoc/guides/users-guide/Histograms.html

Exercise 3

- **1- and 2-dimensional distributions with 3-body system (pi+pi-pi0) from experiment VES**
- **Needed: go to your workdir**
- **ln –s /nfs/lfi.mipt.su/data/nikola/ves/run42/ntbeam_cher_r17_1_v15.root ntbeam_cher_r17_1_v15.root**
- **scp –p /nfs/lfi.mipt.su/data/nikola/ves/run42/my_test_4add.C .**
- **root**
- **.X my_test_4add.C**
- **h_002 -> Draw()**
- **Then fill h_003 and h_004 and h_003->Draw(“box”) or “lego”**