One week introductory course to the software used in high energy physics

by Peter Christiansen, Lund University.

The focus of this week is to make three small projects that will familiarize you with most of the software that you will use for your 10 week project.

The three projects are:
Write a small project description in Latex
Compare PYTHIA simulations to data.
Use the results from the program to write a short report and prepare a presentation

In that way the hope is that you will learn something about:
linux (SUSE) - the operative system we normally use

The linux shell - how to navigate and use shell commands

Emacs - an example of a general purpose editor

Latex - the program you will use to make your report in

C++ - the programming language you will work with

ROOT - the typical C++ analysis framework

PYTHIA – A state of the art pp monte carlo simulation program

ooffice - a free software package similar to MS Office

This is clearly a very ambitious and in some way unrealistic program if you should do it on your own, but if you should start in small steps you would not learn anything useful for your project. Further more, help is also available for the whole week, so you are not jumping right in all alone:-)

Day 1&2: Making your project description

- Linux kernel: one for all systems
  - Linux versions: SuSe, Red hat, Ubuntu
- Introduce command line concept
- Shells - I typically use tcsh
  - .cshrc
- Environment varaibles
  - $HOME
  - $PATH
  - $LD_LIBRARY_PATH
- Basic shell commands (use man to learn about them)
  - ls
  - cat
  - more/less
  - rm
  - cp
  - touch
  - echo
  - man
  - ln
  - mkdir
  - pwd
  - rmdir
  - redirect output | and >
    - shell wizards!

  Exercise 1:
  - login
  - ls -alt vs ls
  - cat .bashrc
  - less/more .bashrc
  - mkdir
  - intro
  - cd intro
  - echo 'hello world'
  - echo 'hello world' > hello.txt
  - ls
  - cp hello.txt hello.txt.bak
  - rm hello.txt

  Exercise 2
  - Add a shortcut to firefox on the desktop
  - Open firefox and google for emacs reference card and then latex cheatsheet
  - Open and print those on printer lw4 (fast black and white printer)

- Emacs – general purpose text free editor
  - open source
    - available for both windows and Linux

- Not graphical like ooffice/word
- Useful for programming and latex and .....
- Give reference card and introduction
- How to find documentation on emacs etc.
  - info
  - google
- How to print
  - Firefox
  - command line: lpr
- Exercise 1:
  - Write a short master project description (project.txt)
    - Follow the reference card
  - Use spell check in emacs to correct the language
    - use also commandline spell check: aspell -c
  - print the files from emacs
    - and from the commandline
- Optional exercise: Study the bigger introduction
  - Write a short cv (cv.txt)

- Latex : a program for making text documents
  - A bit similar to programming language
- input is ACSII file with special latex comments
- Get cheat sheet
- Exercise 2:
  - cp project.txt to project.tex
  - Make a latex file from that following the cheat sheet
  - “compile” with latex
  - convert to ps with dvips
  - display with gv
  - print
  - use also pdflatex
  - Find a picture of the ILC test TPC and insert it in the document
    - You might have to convert it to .eps with the command: convert
  - Label the figure and make a reference to it
  - Find a recent preprint relevant for your project
    - insert and equation from the paper in your document and cite the reference paper
    - How to find papers (find the web pages with google!)
      - spires
      - arxiv
        - Note that you can sometimes also find the figures in the gzip
- Optional exercise: make a nice latex cv from your cv.txt
  - use a table to summarize your education
  - use a bulleted list to list your forces
  - print

Day 3: Introduction to C++/ROOT

- Benefits of object oriented programming
  - Classes and methods
  - Class is “contained”
  - Inheritance
  - Can define smart handling like + for matrixes!
    - matrix a, b, c; c = a+b;
  - define variables and objects where they are needed (and with local scope)
  - Templates (STL)
  - constructor/destructor
- Examples of OO languages
  - C++
  - Java
- What is ROOT?
  - CERN library developed for physics analysis
  - Advantages
    - Many useful classes: Histograms, Functions, Fitting, Graphics
    - Interactive shell-like mode
      - with tab-complete
    - Macros can easily be made into programs (uses standard C++)
    - Very easy input/output with compression
    - Easy to build your own library on top: e.g. AliRoot and brat
  - Explain int -> Int_t
  - Explain basic object TObject
  - Explain pointers and new!
  - web page reference+ examples+manual! Class reference
    - BOOKMARK!!!!!

Exercises:

- Download and install ROOT, see appendix.
- Download from this course web page the program to generate random numbers according to sin(x) (0 < x < pi) in 3 ways
  - box
  - invert
    - explain idea: probability proportional to integral
    - Explain that some methods use both when function can't be inverted
  - ROOT general method for any function
- Exercise
  - Generate uniformly distributed circle (2d hist)
    - Box (calculate pi!)
    - Analytical (how to generate r and theta that the distribution is uniform)

Day 4: Studying PYTHIA simulations

The idea is to study PYTHIA simulations and compare the results to measurements at LHC by ALICE and CMS.

Go to home dir:

```
cd
```

Make pythia dir:

```
mkdir pythia
cd pythia
```

Get the link to the latest PYTHIA version from the webpage:
http://home.thep.lu.se/~torbjorn/Pythia.html

Copy it to your laptop:

```
wget http://home.thep.lu.se/~torbjorn/pythia8/pythia8135.tgz
```

Unpack:

```
tar -xvzf pythia8135.tgz
cd pythia8135
```

Configure PYTHIA for shared libraries

```
./configure --enable-shared
```

Now you have to be hand remove the pedantic options in because this will not work later when we compile with ROOT.

```
emacs config.mk
```

edit, save, and close.
Now you can compile PYTHIA.

```
make
```

To be able to use PYTHIA in our code we need to define/modify some environment variables.
In cshell do:

```
setenv MYPYTHIA ${HOME}/pythia/pythia8135
setenv PYTHIA8DATA ${MYPYTHIA}/xmldoc
setenv LD_LIBRARY_PATH ${MYPYTHIA}/lib:${LD_LIBRARY_PATH}
```
(This you can add to your .cshrc file)

Now we want to make a program that can simulate min. bias collisions and output the data as a tree.
Make a work dir.

```
cd ~/pythia/
mkdir work
cd work
```

Get the Makefile

```
wget http://www.hep.lu.se/staff/christiansen/mastersintro/Makefile
```

Get the program

```
wget
http://www.hep.lu.se/staff/christiansen/mastersintro/charged_particle
_tree.cc
```

Compile the program (NB! name without cc)

```
make charged_particle_tree
```

Execute the program

```
./charged_particle_tree.exe
```

Look at the output in ROOT

```
root
gSystem->Load("libEG");
TFile file("charged_particle_tree.root")
new TBrowser()
```

Exercise:
The charge particle histogram is not normalized. Normalize it properly (remember to recompile the program if you modify it) and compare the result (dN/deta) and compare it to the measurement by ALICE and CMS.

ALICE paper:
http://arxiv.org/abs/0911.5430
CMS paper:
http://arxiv.org/abs/1002.0621

Optional work:
Use the tree to draw and study different observables.
In $MYPTHIA/examples you can find many more advanced PYTHIA examples.

Day 5: Make a short report and a presentation

Write report and make presentation

- Finish simulation and make report in latex with figures (max 2 pages text) and a presentation in ooffice (max 10 pages)
- Some suggested plots could be
  - An example of a cluster
  - The space point resolution
    - For a given drift length/track angle

Appendix: Installing ROOT

Use YAST to install the following packages:
mesa-devel
x11-devel
xml2-devel
subversion

Open a new terminal:

```
cd $HOME
mkdir software
cd software
```

Download ROOT:

```
svn co https://root.cern.ch/svn/root/tags/v5-26-00 root
```

Comment: v5-26-00 specifies the ROOT version (v. 5.26.00). You can look at the ROOT web page to find the latest greatest version. You have to, at least temporally, accept the connection to the ROOT web site.
ROOT takes some time to download so meanwhile you can add the following lines to your .cshrc file:

```
setenv ROOTSYS $HOME/software/root
setenv PATH ${ROOTSYS}/bin:$PATH
if ($?LD_LIBRARY_PATH) then
    setenv LD_LIBRARY_PATH ${ROOTSYS}/lib:$LD_LIBRARY_PATH
else
    setenv LD_LIBRARY_PATH ${ROOTSYS}/lib
endif
```

When you have downloaded ROOT.

```
cd root
./configure --with-f77=gfortran
make
```

And then you have to wait a long time.
When ROOT finishes to compile you can check that it works.
Open a new terminal and execute `root`
This should start up the ROOT interactive shell.
Now you are ready for trying it out.