Introduction to Linux and C++

1. Introduction to Linux

A Linux terminal is a text interface that allows you to write and execute commands. The objective of this task is to explore some of the simplest and more commonly used Linux commands.

1. Open a terminal (shell);
2. Create a directory called *test*: `mkdir test`
3. Enter into the directory: `cd test`
4. Start a text editor to write a C++ program (such as, for example, edit, vim ou emacs): `gedit myfile.txt`
5. Write something, save the file and exit.

Other useful Linux commands:

Command	**Utility**
pwd	Shows the path of the current directory
mkdir *name*	Creates the directory called “name”
cd *name*	Enters into the directory “name”
gedit	Opens the text editor
ls	Lists the contents of the current directory (files and directories)
ls –l	Same as before but more information in the list
cp, mv	Copies or moves files
man *command*	Opens the manual of the “command”
command -h	Prints help with respect to the command syntax
g++	Compiles programs in C and C++
command &	Executes a command in background
exit	Closes the terminal

2. Introduction to C++

The objective of of this task is to write your first C++ program.

1. Open a terminal (shell);
2. Create a directory: `mkdir hello`
3. Enter into that directory: `cd hello`
4. Start an editor: `gedit hello.cpp`
5. Write the following code::

```
#include <iostream>
using namespace std;

int main ( )
{
   cout << “Hello World.” << endl;
   return 0;
}
```

6. Compile your code: `g++ hello.cpp –o hello.x`
7. Run your code: `./hello.x`
8. Modify the output text, compile and execute again your program.

3. Some more simple examples in C++

Program to generate a table with the values of a parabolic function (parabola.cpp).

```
#include <iostream>
using namespace std;

int main ( )
{
   for (int i=1; i<=10; i++)
   {
      cout << i << “\t” << i*i << endl; //prints the value of i and its squared
   }
   return 0;
}
```

Run this program saving the output information in a file called *parabola.dat*:

```
./parabola.x > parabola.dat
```

Later on you will be able to draw a plot with these data using ROOT.

4. Simple array

Implement a program that defines an *array* with the following values:

{10.5, 9.3, 11.4, 10.9, 13, 8.4, 9.2, 8.9, 10.3, 11.2, 12.1, 8.4, 9.2, 9.9, 10.1}

The program should go over all the values and print them in the screen. It should then ask the user to introduce a number between 1 and 15 and print the corresponding element of the array on the screen.

5. Calculate average values and standard deviations

Modify the previous program to calculate the following quantities:

1. Average of the listed numbers

$$\langle \boldsymbol{x} \rangle = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i$$

2. Variance:

$$Var = \frac{1}{N} \sum_{i=1}^{N} (x_i - \ x\rangle)^2$$

3. Standard deviation: $\sigma = \sqrt{Var}$

6. Matrices (two dimensional arrays)

Write a program that calculates the product of the two following matrices and prints the result:

$$\begin{pmatrix} 2 & 3 & 4 \\ 1 & 4 & 9 \\ 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 2 & 1 & 1 \\ -1 & 2 & 1 \\ -1 & -1 & -2 \end{pmatrix}$$