

Introdução ao Linux e ao C++

1. Introdução ao Linux

Um terminal em Linux é um interface de texto que permite escrever e executar comandos. O objetivo desta tarefa é explorar alguns dos comandos mais simples.

1. Abra um terminal (shell);
2. Crie uma nova diretoria *teste*: `mkdir teste`
3. Entre na diretoria: `cd teste`
4. Inicie um software de edição de texto para escrever o programa em C++ (por exemplo, `gedit`, `vim` ou `emacs`): `gedit test.txt`
5. Escreva qualquer coisa, grave o ficheiro e saia do programa.

Outros comandos úteis de Linux são:

Comando	Utilidade
<code>pwd</code>	mostrar o caminho para a diretoria atual
<code>mkdir name</code>	criar a diretoria “name”
<code>cd name</code>	entrar na diretoria “name”
<code>gedit</code>	entrar no editor de texto
<code>ls</code>	listar os ficheiros e diretorias
<code>ls -l</code>	incluir mais informação na listagem
<code>cp, mv</code>	copiar e mover ficheiros/diretorias
<code>man comando</code>	abrir o manual de um comando
<code>comando -h</code>	ajuda na sintaxe de um comando
<code>g++</code>	compilar programas em C e C++
<code>comando &</code>	executar um comando em segundo plano
<code>exit</code>	fechar o terminal

2. Introdução ao C++

O objetivo desta tarefa é escrever um primeiro código em C++.

1. Abra um terminal (shell);
2. Crie uma nova diretoria *hello*: `mkdir hello`
3. Entre na diretoria: `cd hello`
4. Inicie um software de edição de texto para escrever o programa em C++ (por exemplo, `gedit`, `vim` ou `emacs`): `gedit hello.cpp`
5. Escreva o seguinte código:

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

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

6. Compile o código: `g++ hello.cpp -o hello.x`
7. Corra o código: `./hello.x`
8. Modifique o texto de saída, compile e corra o programa de novo.

3. Mais alguns exemplos simples em C++

Programa para gerar uma tabela com os valores dados por uma parábola (parabola.cpp).

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

int main ( )
{
    for (int i=1; i<=10; i++)
    {
        cout << i << "\t" << i*i << endl; //imprime o valor de i e do seu quadrado
    }
    return 0;
}
```

Corra o programa guardando a informação de saída para um ficheiro parabola.dat :
`./parabola.x > parabola.dat`

Mais adiante poderá desenhar o gráfico usando ROOT.

4. Arrays simples

Implemente um programa simples que defina um *array* com os seguintes valores:

{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}

O programa deve correr todos os valores e imprimi-los para o ecrã. Depois pedirá que o utilizador introduza um número entre 1 e 15 e imprimirá o elemento correspondente para o ecrã.

5. Cálculo de valores medios e desvio padrão

Modifique o programa anterior para calcular as seguintes quantidades:

1. Valor médio dos números da lista $\langle x \rangle = \frac{1}{N} \sum_{i=1}^N x_i$

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

3. Desvio padrão: $\sigma = \sqrt{Var}$

6. Matrizes (arrays de duas dimensões)

Escreva um programa que faça o produto das seguintes matrizes:

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