

A series of white and red curved lines of varying thicknesses and radii, some forming partial circles, arranged in an arc across the top of the image.

Insper **Supercomputação**

A series of white and red curved lines of varying thicknesses and radii, some forming partial circles, arranged in an arc across the bottom of the image.

Reflexão da aula passada

- O que pode causar tempos diferentes rodando o mesmo código?
 - Partição dos dados desequilibrada
 - Overhead de SLURM
 - Carregamento Dinâmico e Inicialização de Bibliotecas
 - *Contention* de Recursos (ambiente compartilhado)
 - Latência de Comunicação

Como pedir? O que pedir?

- Por meio de configurações do arquivo `.slurm`

```
#!/bin/bash
#SBATCH --job-name=convolucao_buffado
#SBATCH --output=buffado_%j.txt
#SUBATCH --ntasks=4
#SBATCH --cpus-per-task=4
#SBATCH --mem=1024
#SBATCH --time=00:10:00
#SBATCH --partition=espec
```



Aula 04

- Programação em C++

Algoritmo

- Sequência finita de passos executáveis que resolve um problema
- Implementar um algoritmo:
 - Transformação de um algoritmo em um programa executável

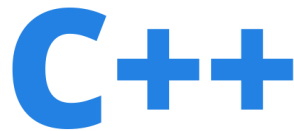
Quanto tempo um programa demora?

- Algoritmo
 - Complexidade computacional (classe de algoritmos)
 - Estruturas de Dados
- Implementação
 - Medido em segundos, para uma certa entrada
 - Tecnologias usadas (linguagens de programação, bibliotecas)
 - Hardware utilizado (clock de CPU, RAM, cache, # de núcleos, etc.)

Quanto tempo um programa demora?

- Supercomputação começa quando desafios de programação acaba
- Dado um “bom” algoritmo, vamos definir:
 - Linguagem de programação adequada
 - Paralelismo indicado
 - Implementação paralela eficiente

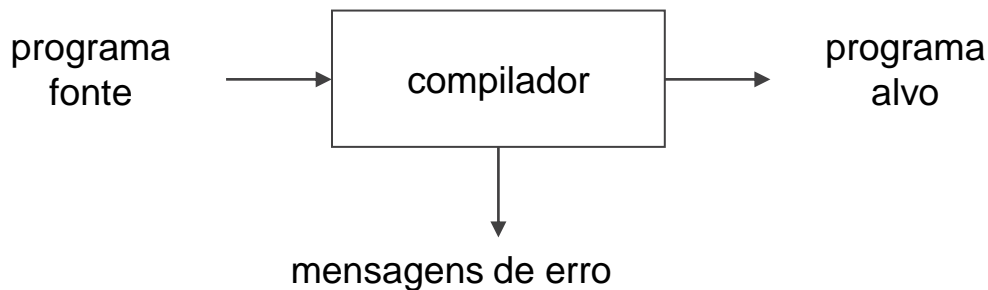
Linguagem C++



- Linguagem de programação criada no Bell Labs, em 1985
- É uma das linguagens de programação mais usadas no mundo
- C++ é derivado da linguagem C, contudo possui uma série de recursos adicionais que traz grandes vantagens para o desenvolvedor

O processo de compilação

- Compilador é um software que lê a especificação de um programa em uma linguagem-fonte o traduz em um programa em uma linguagem-alvo



O processo de compilação

- Para compilar um programa em C++ você pode usar a chamada g++

```
$ g++ your_file.cpp -o your_program
```

- Se você estiver no MacOS, uma possível linha para compilar é:

```
$ clang++ -std=c++11 -stdlib=libc++ -Wall hello.cpp -o hello
```

C++, Hello World

- Um programa "Hello World" em C++ é muito parecido com um em C.

```
#include <iostream>

int main() {

    std::cout << "Hello World!\n";

}
```

Pontos comuns com C

- Função principal: `int main(int argc, char *argv[]);`
- Comentários: `/* */` `//`
- Tipos de dados: `int`, `float`, `double`, `char`
- Variações de dados: `unsigned`, `short`, `long`
- Qualificadores de variáveis: `const`, `static`
- Casting: `(int)` `(float)` `(char)`
- Operações: `+`, `-`, `*`, `/`, `%`
- Atribuição: `=`, `+=`, `-=`, `*=`, `/=`, `%=`, `>>=`, `<<=`, `&=`, `^=`, `|=`
- Incremento e decremento: `++` `--`
- Operadores lógicos: `!`, `&&`, `||`
- Operador condicional ternário : `(? :)`
- Operador vírgula: `(,)`

Pontos comuns com C

- Operadores Bitwise : (&, |, ^, ~, <<, >>)
- Construção Condicional : if, else, switch
- Loops: while, do-while, for
- Comparadores: ==, !=, >, <, >=, <=
- Diretivas de pré-processamento (#define, etc.)
- Declaração de funções: type func(...) { ... }
- Vetores e Matrizes: type name [elements][...];
- Enumeradores: enum
- Organização de dados: structs
- Redefinidor de tipos: typedef

C++ - Input/Output (Streams)

- C++ utiliza uma abstração conveniente denominada Streams para executar a entrada e saída de dados.

stream	description
cin	standard input stream
cout	standard output stream
cerr	standard error (output) stream
clog	standard logging (output) stream

C++ - Saída de dados (iostream)

- Utilize o `std::cout` com dois sinais de menor (`<<`) para indicar que quer enviar os dados para o console.

```
std::cout << "Bom dia" << std::endl;
```

- `std::cout` aceita qualquer tipo de dados.

C++ - Entrada de dados (iostream)

- Use o `std::cin` com dois sinais de maior (`>>`) para indicar que quer capturar os dados do console

```
string texto;
```

```
std::cin >> texto;
```


C++ - Namespaces

- O namespace declara uma região de escopo para os identificadores
- Exemplo de uso de namespace:

```
std::cout << "Ola";
```

- Outra possibilidade:

```
using namespace std;
```

```
cout << "Ola";
```

C++ - Inicialização de variáveis

- Tradicionalmente

```
type identifier = initial_value;
```

- Inicialização com valor inicial

```
type identifier (initial_value);  
int x (0);
```

- Ou

```
type identifier {initial_value};  
int x {0};
```

C++ - Inicialização Vetores/Matrizes

- C++ permite preencher um vetor sem definir seu tamanho, se você deixar os colchetes vazios []
- Nesse caso, o compilador assumirá automaticamente o tamanho para o vetor que corresponde ao número de valores incluídos entre as chaves { }.

```
int foo [] = { 16, 2, 77, 40, 12071 };
```

Tipos de dados

- Principais novos tipos de dados que foram introduzidos em C++
 - bool: true / false
 - string: armazena textos e possui recursos para tratar textos

```
#include <string>
```

```
string texto;
```

```
texto = "exemplo de texto";
```

```
std::cout << texto << std::endl;
```



Atividade prática

Nesta aula iremos implementar códigos em C++

Todas serão implementações sequenciais (por enquanto)

O que pedir?

- Temos apenas código sequenciais, logo nosso arquivo *.slurm* deve ser:

```
#!/bin/bash
#SBATCH --job-name=convolucao_buffado
#SBATCH --output=buffado_%j.txt
#SBATCH --ntasks=____
#SBATCH --cpus-per-task=____
#SBATCH --mem=1024
#SBATCH --time=00:10:00
#SBATCH --partition=espec
```

Atividades

- 1. Olá Mundo (Hello World):** Escreva um programa simples que exibe a mensagem "Olá, Mundo!" na tela.
- 2. Calculadora Simples:** Crie um programa que peça ao usuário para inserir dois números e uma operação (adição, subtração, multiplicação ou divisão). O programa deve exibir o resultado da operação.
- 3. Verificação de Número Par ou Ímpar:** Escreva um programa que peça ao usuário para inserir um número e informe se ele é par ou ímpar.
- 4. Contagem de Letras em uma String:** Escreva um programa que peça ao usuário para inserir uma string e exiba o número de caracteres na string.
- 5. Soma de Números em um Vetor:** Crie um programa que solicite ao usuário para inserir 5 números em um vetor e depois exiba a soma desses números.
- 6. Encontrar o Maior Número em um Vetor:** Escreva um programa que peça ao usuário para inserir 10 números em um vetor e determine qual é o maior número.
- 7. Ordenação de Vetor (Bubble Sort):** Implemente o algoritmo de ordenação Bubble Sort para organizar um vetor de números em ordem crescente.
- 8. Matriz e Soma de Elementos Diagonais:** Crie um programa que peça ao usuário para inserir uma matriz 3x3 e depois exiba a soma dos elementos na diagonal principal.
- 9. Manipulação de Strings: Palíndromo:** Escreva um programa que verifique se uma string inserida pelo usuário é um palíndromo (lê-se da mesma forma de trás para frente).
- 10. Sistema de Controle de Estoque Simples:** Crie um programa que simule um sistema de controle de estoque para uma loja. O programa deve permitir ao usuário adicionar itens, remover itens, e visualizar o estoque atual. Os dados devem ser armazenados em vetores.
- 11. Leitura e Escrita em Arquivo:** Estatísticas de Texto: Escreva um programa que leia um arquivo de texto e calcule as seguintes estatísticas: número total de palavras, número total de linhas, número médio de palavras por linha e a palavra mais frequente. Em seguida, grave essas estatísticas em um novo arquivo de texto.

Obrigado