



FAETERJ

Disciplina: GMMSM - C++

Aaron Leão

e-mail: aaron@lncc.br

Data: 3 de dezembro de 2018

Flags de compilação, standard

1. Como vimos, o C++ está em constante evolução e possui algumas versões desde que foi integrado ao ANSI. As versões e como compilar utilizando elas.

- (a) (1998): `-std=c++98`
- (b) (2003): `-std=c++03`
- (c) (2011): `-std=c++11`
- (d) (2014): `-std=c++14`
- (e) (2017): `-std=c++17`
- (f) (2020): `-std=c++20`

Números inteiros em C possui limite inferior mas não possuem limites superior. Para manter retrocompatibilidade com C, o C++ mantém a keyword `int` efetuando a mesma coisa que o C. Para se limitar o tamanho em bytes que um número inteiro deve utilizar usamos as seguintes keywords.

1. `int8_t`, `uint8_t`
2. `int16_t`, `uint16_t`
3. `int32_t`, `uint32_t`
4. `int64_t`, `uint64_t`

No código abaixo possuímos algumas coisas peculiares das versões. Na linha 3 temos a prototipação de uma função recursiva para o cálculo da sequência de Fibonacci, utilizando as keywords `constexpr` e `uint64_t` para isso.

```
1  #include <iostream>
2
3  constexpr uint64_t fib(int n)
4  {
5      if(n<=1)
6          return 1;
7      return fib(n-1) + fib(n-2);
8  }
9
10 int main(int argc, char** argv)
11 {
12     int n = atoi(argv[1]);
13     const uint64_t result = fib(50);
14     const uint64_t result = fib(n);
15     std::cout << "fib de " << n << ": " << result << '\n';
16 }
```

Figura 1: Comentar a linha 13 e compilar. Futuramente descomentar a linha 13 comentar a linha 14.

Primeiramente comentar a linha 13 e compilar. Futuramente descomentar a linha 13 comentar a linha 14. Para entender rapidamente como algumas funcionalidades foram adicionadas no C++ ao longo do tempo. Pode-se efetuar o teste compilando o código abaixo com o os padrões definidos acima.

```
$ g++ -std=c++98 encontro1.cpp, não reconhece constexpr, atoi, uint64_t
$ g++ -std=c++03 encontro1.cpp, não reconhece constexpr, atoi, uint64_t
$ g++ -std=c++11 encontro1.cpp, não reconhece constexpr
$ g++ -std=c++14 encontro1.cpp, compilado sem erro
$ g++ -std=c++17 encontro1.cpp, compilado sem erro
```

Se observar os erros vão desaparecendo. Observem que para executar precisa de um parâmetro que é o número de Fibonacci que será calculado.

```
$ ./a.out 10
```

Meça o tempo de execução de um Fibonacci grande como por exemplo 45.

```
$time ./a.out 10
```

Agora descomente a linha 13 e comente a linha 14, compile e rode.

```
$ g++ -std=c++17 encontro1.cpp
$time ./a.out 10
```