

Linguagem de Programação II

Prof. Antonio Carlos Sobieranski

DEC7532 | ENC | DEC | CTS



UNIVERSIDADE FEDERAL
DE SANTA CATARINA

Unidade 01 – Introdução ao C++

Parte 3c. Depuração de Código

- gdb
- gdb com codeblocks

Parte 3d. Novos elementos do C++:

- `std::stringstream`
- `std::pair`
- `std::tuple`

Unidade 01 – Introdução ao C++

gdb (GNU Project Debugger)

- O GDB (*GNU Project Debugger*) é uma ferramenta para:
 - observar um programa enquanto este executa
 - ver o estado no momento que a execução falha
- Permite:
 - iniciar a execução de um programa
 - executar linha-a-linha
 - especificar pontos de paragem
 - imprimir valores de variáveis
- Suporta C, C++, Objective-C, Ada e Pascal (entre outras linguagens)

Unidade 01 – Introdução ao C++

`gdb` (*GNU Project Debugger*) – Terminal / Linha de Comando

```
g++ -g main.cpp -o exec
```

Flags:

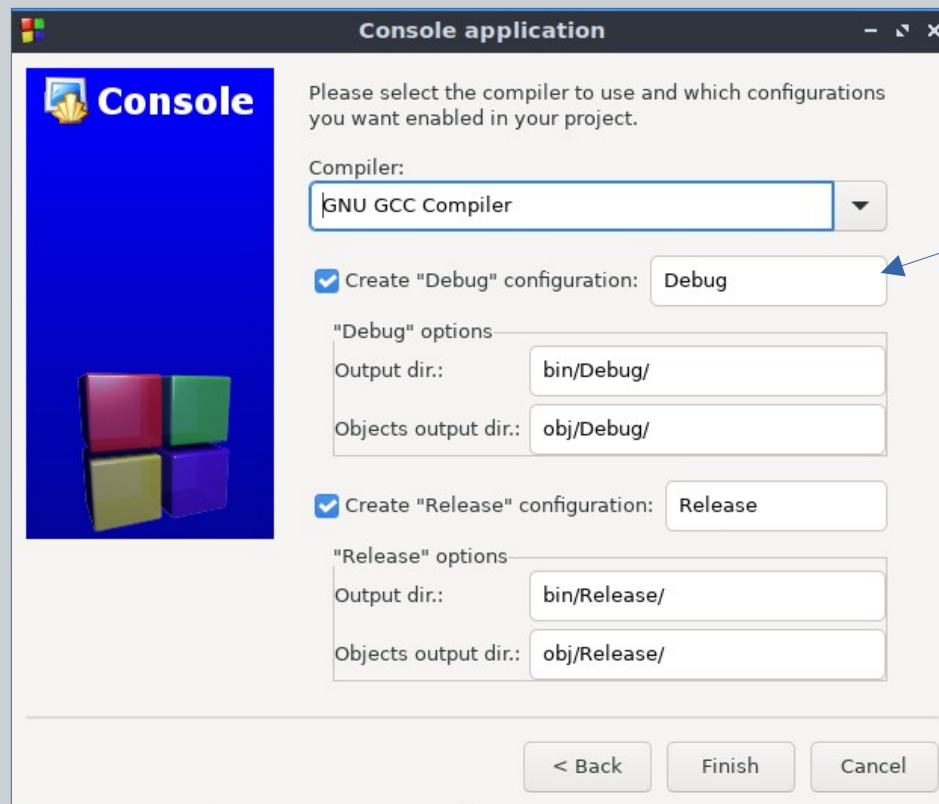
-g Builds executable with debugging symbols for GDB GNU Debugger or Clang Debugger. It should only be used during development for debugging builds.

```
gdb ./exec
```

<i>b</i>	<i>breakpoint</i>	<i>s</i>	<i>step into</i>
<i>r</i>	<i>run</i>	<i>p</i>	<i>print</i>
<i>c</i>	<i>continue</i>	<i>bt</i>	<i>backtracking</i>
<i>n</i>	<i>next</i>		

Unidade 01 – Introdução ao C++

gdb (GNU Project Debugger) – no Codeblocks



Unidade 01 – Introdução ao C++



gdb (GNU Project Debugger) – no Codeblocks

The screenshot shows the Code::Blocks IDE with a C++ program in `main.cpp`. The program is as follows:

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main()
7  {
8      cout << "Hello world!" << endl;
9
10     int v;
11     v = 10;
12
13     vector<int> listInts;
14     listInts.push_back(100);
15
16     cout << "Done" << endl;
17
18     return 0;
19 }
20
```

The Watches window is open, showing the following data:

Watches	
Function argument	
Locals	
v	10
listInts	std::vector of length
[0]	100

Unidade 01 – Introdução ao C++

Parte 3d. Novos elementos do C++: *std::stringstream*

```
#include <sstream>
```

Objetos desse tipo usam um *buffer* de *strings* que contém cadeias de caracteres.

Caracteres podem ser inseridos ou extraídos da cadeia usando qualquer operador relacionado.

```
stringstream sst;  
sst << "Olá pessoas. ";  
sst << "Testando o stringstream. Muito versátil e útil para coletar";  
sst << " ";  
sst << ", manipular e gerenciar texto na linguagem C";  
sst << "++" << endl;  
  
cout << sst.str() << endl << sst.str().size() << endl;  
cout << sst.str().at(0) << endl;
```

Unidade 01 – Introdução ao C++

Parte 3d. Novos elementos do C++: *std::stringstream*

Exercício 1:

1. Ler um arquivo de texto com ***ifstream***, caractere a caractere, e armazenar com ***stringstream***.

Usar os arquivos de manuscrito contidos no Moodle.

Leitura caractere a caractere: (***fileReaderFreq*** é do tipo ***ifstream***)

```
char ch;  
while(fileReaderFreq.get(ch))
```

2. Substituir todos os caracteres do texto por outro de sua escolha.

Unidade 01 – Introdução ao C++

Parte 3d. Novos elementos do C++: *std::pair*

#include <utility>

Constrói um par de objetos de qualquer tipo, inclusive pares de pares.

```
pair< int, int> p1;  
p1.first = 10;  
p1.second = 20;  
cout << p1.first << " -> " << p1.second << endl;
```

Outras formas de inicialização:

```
int main () {  
    std::pair <std::string, double> product1;           // default constructor  
    std::pair <std::string, double> product2 ("tomatoes", 2.30); // value init  
    std::pair <std::string, double> product3 (product2); // copy constructor  
  
    product1 = std::make_pair(std::string("lightbulbs"), 0.99); // using make_pair (move)  
  
    product2.first = "shoes"; // the type of first is string  
    product2.second = 39.90; // the type of second is double  
  
    std::cout << "The price of " << product1.first << " is $" << product1.second << '\n';  
    std::cout << "The price of " << product2.first << " is $" << product2.second << '\n';  
    std::cout << "The price of " << product3.first << " is $" << product3.second << '\n';  
    return 0;  
}
```

Unidade 01 – Introdução ao C++

Parte 3d. Novos elementos do C++:

#include <tuple>

Uma tupla é um objeto capaz de gerenciar uma coleção de objetos. Cada elemento pode ser de tipos distintos.

```
1 // tuple example
2 #include <iostream>      // std::cout
3 #include <tuple>         // std::tuple, std::get, std::tie, std::ignore
4
5 int main ()
6 {
7     std::tuple<int,char> foo (10,'x');
8     auto bar = std::make_tuple ("test", 3.1, 14, 'y');
9
10    std::get<2>(bar) = 100;                // access element
11
12    int myint; char mychar;
13
14    std::tie (myint, mychar) = foo;          // unpack elements
15    std::tie (std::ignore, std::ignore, myint, mychar) = bar; // unpack (with ignore)
16
17    mychar = std::get<3>(bar);
18
19    std::get<0>(foo) = std::get<2>(bar);
20    std::get<1>(foo) = mychar;
21
22    std::cout << "foo contains: ";
23    std::cout << std::get<0>(foo) << ' ';
24    std::cout << std::get<1>(foo) << '\n';
25
26    return 0;
27 }
```

Unidade 01 – Introdução ao C++

Parte 3d. Novos elementos do C++:

Exercício 2:

1. Ler um arquivo de texto com ***ifstream***, caractere a caractere, e armazenar com ***stringstream***.

Usar os arquivos de manuscrito contidos no Moodle.

a) Implementar com ***std::pair*** ou ***std::tuple*** um algoritmo que forneça a quantidade de caracteres de cada tipo.

b) Apresentar uma lista em ordem decrescente pelo número de caracteres.

Contato

Prof. Antonio Carlos Sobieranski – DEC | A316JD / 206MA

E-mail: a.sobieranski@ufsc.br

<https://lsim.ufsc.br>



UNIVERSIDADE FEDERAL
DE SANTA CATARINA