

Comparando C/C++ com Assembly

TIME



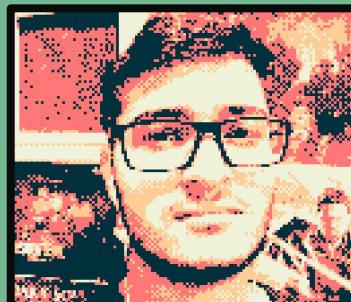
Gabriel Teixeira



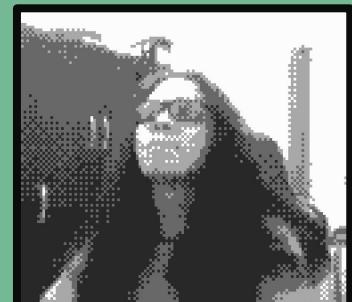
Helder Schramm



Natan Santos



Victor Yan



Vinicius Soares

Programa 1 - Intervalo entre dois números inteiros (C/C++)

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int num1, num2;

    printf("Digite o primeiro numero: ");
    scanf("%d", &num1);
    printf("Digite o segundo numero: ");
    scanf("%d", &num2);

    if (num1 > num2)
        num1 = num2;

    for (int i = num1; i < num2; i++)
    {
        printf("%d ", i);
    }
    return 0;
}
```

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

int main()
{
    int num1, num2;

    cout << "Digite o primeiro numero: ";
    cin >> num1;
    cout << "Digite o segundo numero: ";
    cin >> num2;

    if (num1 > num2)
        num1 = num2;

    for (int i = num1; i < num2; i++)
    {
        printf("%d ", i);
    }
    return 0;
}
```

Programa 1 - Intervalo entre dois números inteiros (Assembly)

```
1 ;nasm -felf64 code.asm && gcc code.o && ./a.out
2
3 ▾ SECTION .data
4     message1: db "Digite o primeiro numero: ",0
5     message1Len: equ $-message1
6     message2: db "Digite o segundo numero: ",0
7     message2Len: equ $-message1
8     message3: db "Os numeros sao:", 0
9     formatin: db "%d",0
10    formatout: db "%d",10,0
11    integer_1: times 4 db 0
12    integer_2: times 4 db 0
13
14 ▾ SECTION .text
15     global _start
16     global main
17     extern scanf
18     extern printf
19
20 ▾ main:
21     push rbx
22
23     ; Print msg1
24     mov eax, 4
25     mov ebx, 1
26     mov ecx, message1
27     mov edx, message1Len
28     int 80h
29
30     ; Scans num1
31     mov rdi, formatin
32     mov rsi, integer_1
33     mov al, 0
34     call scanf
35
```

```
35
36     ; Print msg2
37     mov eax, 4
38     mov ebx, 1
39     mov ecx, message1
40     mov edx, message1Len
41     int 80h
42
43     ; Scanf num2
44     mov rdi, formatin
45     mov rsi, integer_2
46     mov al, 0
47     call scanf
48
49     ; Loop de num1 ate num2
50     mov ebx, [integer_1]
51     loop:
52         mov rdi, formatout
53         mov esi, ebx
54         xor eax, eax
55         call printf
56         inc ebx
57         cmp ebx, [integer_2]
58         jle loop
59
60     pop rbx
61     mov rax,0
62
63     ret
```

Programa 2 - Maior número

```
#include <stdio.h>

int main(){
    // declarando variáveis que armazenarão os dois números e o maior
    int a, b, maior;

    // Recebendo os dois números do usuário e armazenando nas variáveis "a" e "b"
    scanf("%d", &a);
    scanf("%d", &b);

    // Verificando se "a" é o maior número
    if(a > b){

        // "maior" recebe o valor de "a"
        maior = a;
        // imprimindo o valor de "maior"
        printf("%d ", maior);
    }

    // Verificando se "b" é o maior número
    else if(b > a){

        // "maior" recebe o valor de "b"
        maior = b;
        // imprimindo o valor de "maior"
        printf("%d ", maior);
    }

    // Se nenhum dos números for maior nem menor
    else{

        // Imprime "a" (os dois são iguais mesmo)
        printf("%d\n ", a);
        // Imprime a mensagem abaixo
        printf("Os numeros são iguais");
    }

    printf("\n");
}

return 0;
}
```

C

```
#include <iostream>

using namespace std;

int main(){

    // declarando variáveis que armazenarão os dois números e o maior
    int a, b, maior;

    // Recebendo os dois números do usuário e armazenando nas variáveis "a" e "b"
    cin >> a;
    cin >> b;

    // Verificando se "a" é o maior número
    if(a > b){

        // "maior" recebe o valor de "a"
        maior = a;
        // imprimindo o valor de "maior"
        cout << maior << "\n";
    }

    // Verificando se "b" é o maior número
    else if(b > a){

        // "maior" recebe o valor de "b"
        maior = b;
        // imprimindo o valor de "maior"
        cout << maior << "\n";
    }

    // Se nenhum dos números for maior nem menor
    else{

        // Imprime "a" (os dois são iguais mesmo)
        cout << a;
        // Imprime a mensagem abaixo
        cout << "Os números são iguais" << "\n";
    }

}

return 0;
}
```

C++

Programa 2 - Imprimir o maior de 2 números (Assembly)

```
1 ;nasm -felf64 code.asm && gcc code.o && ./a.out
2
3 SECTION .data
4     message1: db "Primeiro numero: ",0
5     message1Len: equ $-message1
6     message2: db "Segundo numero: ",0
7     message2Len: equ $-message2
8     message3: db "Os numeros sao:", 0
9     formatin: db "%d",0
10    formatout: db "Maior número: %d",10,0
11    integer_1: times 4 db 0
12    integer_2: times 4 db 0
13
14 SECTION .text
15     global _start
16     global main
17     extern scanf
18     extern printf
19
20 main:
21     push rbx
22
23     mov eax, 4
24     mov ebx, 1
25     mov ecx, message1
26     mov edx, message1Len
27     int 80h
28
29     mov rdi, formatin
30     mov rsi, integer_1
31     mov al, 0
32     call scanf
33
```

```
34     mov eax, 4
35     mov ebx, 1
36     mov ecx, message2
37     mov edx, message2Len
38     int 80h
39
40     mov rdi, formatin
41     mov rsi, integer_2
42     mov al, 0
43     call scanf
44
45     mov ebx, [integer_1]
46     cmp ebx, [integer_2]
47     jge resultado
48
49 maior2:
50     mov ebx, [integer_2]
51
52 resultado:
53     mov rdi, formatout
54     mov esi, ebx
55     xor eax, eax
56     call printf
57     inc ebx
58
59 fim:
60     pop rbx
61     mov rax,0
62
63 ret
```

Programa 3 - Fahrenheit para Celsius (C/C++)

```
#include <stdio.h>
#include <stdlib.h>

// função que converte de fahrenheit para celsius
float toCelsius(int degrees){
    return (5.0 * ((degrees-32.0) / 9.0));
}

// função principal
int main(){
    // declarando variáveis
    float fahrenheit;

    // solicitando entrada em formato fahrenheit
    printf("\nGraus fahrenheit: ");
    scanf("%f", &fahrenheit);
    // limpando tela e reescrevendo a entrada de forma organizada
    system("clear");
    printf("\nGraus fahrenheit: %.2f°F", fahrenheit);
    // imprimindo valor convertido com precisão 2
    printf("\nGraus Celsius : %.2f°C", toCelsius(fahrenheit));

    return 0;
}

// EOF
```

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

// função que converte de fahrenheit para celsius
float toCelsius(int degrees){
    return (5.0 * ((degrees-32.0) / 9.0));
}

// função principal
int main(){
    // declarando variáveis
    float fahrenheit;

    // solicitando entrada em formato fahrenheit
    cout << "\nGraus fahrenheit: ";
    cin >> fahrenheit;
    // limpando tela e reescrevendo a entrada de forma organizada
    system("clear");
    cout << "\nGraus fahrenheit: " << fixed << setprecision(2) << fahrenheit;
    // imprimindo valor convertido com precisão 2
    cout << "\nGraus Celsius : " << fixed << setprecision(2) << toCelsius(fahrenheit);

    return 0;
}

// EOF
```

Programa 3 - Fahrenheit para Celsius (Assembly)

```
1 ; nasm -f elf64 main.asm && gcc -no-pie main.o && ./a.out
2     global main
3     extern printf
4     extern scanf
5     extern puts
6
7 section .text
8 main:
9     ; exibe msg fahrenheit
10    mov rdi, msgFahrenheit
11    call puts
12
13    ; scanf
14    mov rdi, formatin
15    mov rax, 1
16    sub rsp,8
17    call scanf
18    add rsp,8
19    movsd [fahrenheit], xmm0
20
21    ; soma fahrenheit + ap2 e / por 2
22    movsd xmm0, [fahrenheit]
23    subsd xmm0, [_32]
24    movsd xmm1, [_9]
25    divsd xmm0, xmm1
26    movsd xmm1, [_5]
27    mulsd xmm0, xmm1
28    movsd [celsius], xmm0
```

```
29 fim:
30     mov rdi, msgCelsius
31     movsd xmm0, [celsius]
32     mov rax, 1
33     sub rsp, 8
34     call printf
35     add rsp, 8
36     ret
37
38
39
40 section .data
41     msgFahrenheit: db "Temperatura em Fahrenheit", 0
42     msgCelsius: db "Celsius: %.2f",10, 0
43     formatin: db "%lf", 0
44     fahrenheit: dq 0.0
45     pf: dq 0.0
46     celsius: dq 0.0
47     _9: dq 9.0
48     _32: dq 32.0
49     _5: dq 5.0
```

Programa 4 - Média da UEA (C/C++)

```
#include <stdio.h>

int main()
{
    //declarando as variáveis da nota 1 e 2 e lendo as entradas
    float p1, p2;
    scanf("%f",&p1);
    scanf("%f",&p2);
    //a nota da média é calculada com a soma das duas notas sobre 2
    float np = (p1 + p2) / 2;
    //se a média é maior que 8 ele imprime Aprovado
    if (np >= 8.0)
    {
        printf("Aprovado: %.2f\n", np);
    }
    else
    {
        //declarando e recebendo a nota final
        float pf;
        scanf("%f",&pf);
        //a média final é calculada com a soma das três notas divididas por 3
        float nota = (p1 + p2 + pf) / 3;
        //se a média final for maior que 6 ele imprime Aprovado
        if (nota >= 6.0)
        {
            printf("Aprovado: %.2f\n", nota);
        }
        //se a média for menor que 6 ele imprime Reprovado
        else
        {
            printf("Reprovado: %.2f\n", nota);
        }
    }
    //termina o programa
    return 0;
}
```

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

int main()
{
    //declarando as variáveis da nota 1 e 2 e lendo as entradas
    float p1, p2;
    cin >> p1 >> p2;
    //a nota da média é calculada com a soma das duas notas sobre 2
    float np = (p1 + p2) / 2;
    //se a média é maior que 8 ele imprime Aprovado
    if (np >= 8.0)
    {
        cout << fixed << setprecision(2) << "Aprovado: " << np << endl;
    }
    else
    {
        //declarando e recebendo a nota final
        float pf;
        cin >> pf;
        //a média final é calculada com a soma das três notas divididas por 3
        float nota = (p1 + p2 + pf) / 3;
        //se a média final for maior que 6 ele imprime Aprovado
        if (nota >= 6.0)
        {
            cout << fixed << setprecision(2) << "Aprovado: " << nota << endl;
        }
        else
        //se a média for menor que 6 ele imprime Reprovado
        {
            cout << fixed << setprecision(2) << "Reprovado: " << nota << endl;
        }
    }
    //termina o programa
    return 0;
}
```

Programa 4 - Média da UEA (Assembly)

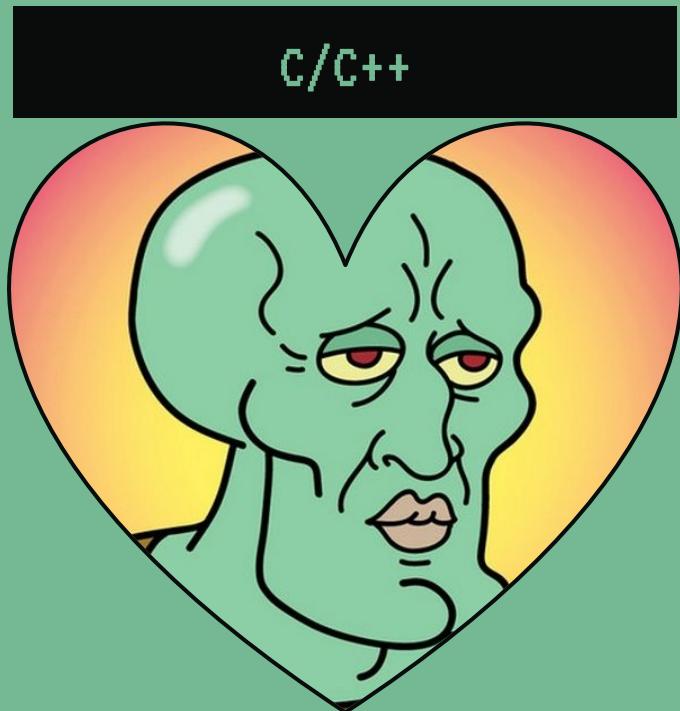
```
1+ ; nasm -f elf64 main.asm && gcc -no-pie main.o && ./a.out
2  global main
3  extern printf
4  extern scanf
5  extern puts
6
7 section .text
8 main:
9   ; exibe msg ap1
10  mov rdi, msgAp1
11  call puts
12
13  ; scanf
14  mov rdi, formatin
15  mov rax, 1
16  sub rsp,8
17  call scanf
18  add rsp,8
19  movsd [ap1], xmm0
20
21  ; msg ap2
22  mov rdi, msgAp2
23  call puts
24
25  ;scanf
26  mov rdi, formatin
27  mov rax, 1
28  sub rsp, 8
29  call scanf
30  add rsp, 8
31  movsd [ap2], xmm0
32
33  ; soma ap1 + ap2 e / por 2
34  movsd xmm0, [ap1]
35  addsd xmm0, [ap2]
36  movsd xmm1, [_2]
37  divsd xmm0, xmm1
38  movsd [media], xmm0
39
40  ;verifica se media >= 8
41  mov rax, [media]
42  cmp rax,[_8]
43  jge aprovado ; pula pra aprovado
44
45  cmp rax, [_4] ; verifica se menor que 4 e ja reprova direto
46  jl reprovado
47
48  mov rdi, msgPF ; msg pf
49  call puts
50
51  mov rdi, formatin ; scanf pf
52  mov rax, 1
53  sub rsp, 8
54  call scanf
```

```
55    add rsp, 8
56    movsd [pf], xmm0
57
58    movsd xmm0, [ap1] ; soma ap1 + ap2 + pf
59    addsd xmm0, [ap2]
60    addsd xmm0, [pf]
61    movsd xmm1, [_3]
62    divsd xmm0, xmm1
63    movsd [media], xmm0
64
65    mov rax, [media] ; verifica se >= 6
66    cmp rax,[_6]
67    jge aprovado
68
69    jmp reprovado ; nao deu
70
71    jmp fim
72
73    aprovado:
74        mov rdi, msgAprovado
75        call puts
76        jmp fim
77
78    reprovado:
79        mov rdi, msgReprovado
80        call puts
81        jmp fim
82
```

```
82
83    fim:
84        mov rdi, msgMedia
85        movsd xmm0, [media]
86        mov rax, 1
87        sub rsp, 8
88        call printf
89        add rsp, 8
90        ret
91
92    section .data
93    msgAp1: db "Digite a nota da AP1", 0
94    msgAp2: db "Digite a nota da Ap2", 0
95    msgPF: db "Digite a nota da PF", 0
96    msgAprovado: db "Aprovado", 0
97    msgReprovado: db "Reprovado", 0
98    msgMedia: db "media: %.2f",10, 0
99    formatin: db "%lf", 0
100   ap1: dq 0.0
101   ap2: dq 0.0
102   pf: dq 0.0
103   media: dq 0.0
104   _2: dq 2.0
105   _3: dq 3.0
106   _4: dq 4.0
107   _6: dq 6.0
108   _8: dq 8.0|
```

Comparando as implementações

Dificuldade



Por serem de linguagens de alto nível, que abstraem as operações com registradores do Assembly, implementar os programas em C ou C++ se torna muito mais fácil do que com Assembly

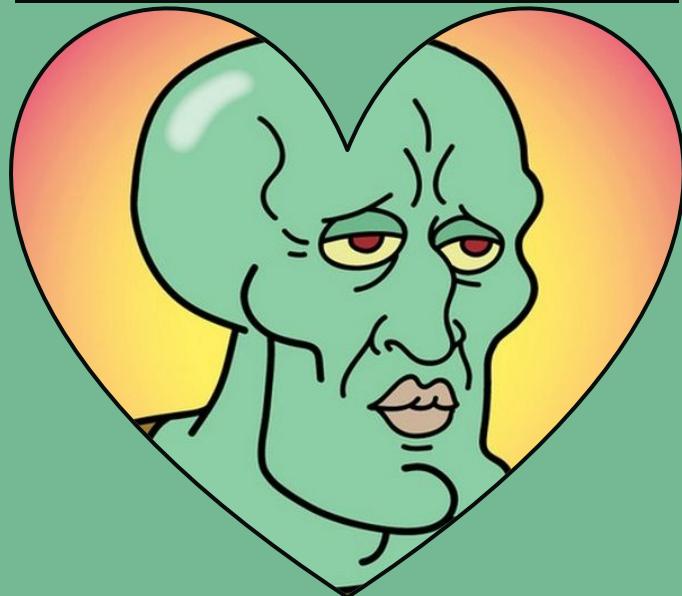
Dificuldade

- Muitas instruções não intuitivas e que não são fáceis de lembrar.
- Muitos registradores, é preciso entender o uso de cada um.
- Necessário lidar com conversões a não ser que externalize com as funções do C.



Legibilidade

C/C++



Mais legível, os nomes das funções são mais fáceis de lembrar

Como as operações com registradores ficam abstraídas, o programador pode focar mais na resolução do problema do que em aspectos técnicos da memória

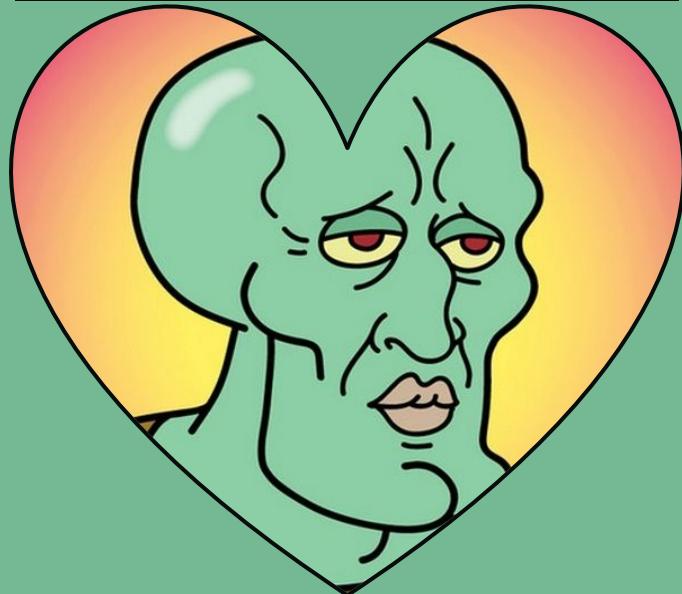
Legibilidade

- É necessário comentar todas as linhas para que fique legível.
- Os registradores por serem padronizados quebram a possibilidade de uma leitura intuitiva do código.
- O código se divide em muitos blocos e são necessárias muitas linhas de comandos para fazer algo simples, o que pode complicar a navegação.



Tamanho do código

C/C++



Por serem linguagens de alto nível, o código fica menor do que em Assembly, operações com 4 ou 5 linhas são realizadas somente em uma.

Tamanho do código

Código muito extenso em relação ao C/C++, pois é preciso manipular com precisão cada registrador para obter os resultados desejados.

Assembly

Miguel Vásquez 3D

