

# Tradução de código em C e comparação com ChatGPT

Marcos Henrique Gollin Filho

1

<sup>2</sup>Ciência da Computação – Pontifícia Universidade Católica de Minas Gerais  
Poços de Caldas, Brasil.

**Resumo.** Este artigo apresenta uma análise comparativa entre códigos Assembly gerados por um modelo de linguagem natural avançado, o GPT, e por programadores humanos. O estudo consistiu na tradução de um código C para Assembly, tanto pelo GPT quanto por programadores humanos. Em seguida, os códigos em assembly foram avaliados em termos de eficiência, legibilidade, tamanho e desempenho.

## 1. Introdução

Com a crescente frequência de uso do ChatGPT e seu impacto na sociedade devido à facilidade de acesso à informação, despertou-se o interesse por um estudo na área da computação. No entanto, é importante notar que a IA generativa, como o ChatGPT, utiliza a internet como fonte de dados de estudo, o que pode resultar em respostas com equívocos. Esta pesquisa busca compreender e distinguir a tradução de um código em C para assembly, estabelecendo uma comparação entre o desempenho de um estudante humano e o do ChatGPT..

## 2. Código em C

Foi utilizado um código fatorial recursivo na linguagem C para ser traduzido.

```
1 #include<stdio.h>
2 int fatorial(int n){
3     if(n <= 1) return 1;
4     return n * fatorial(n-1);
5 }
6
7 int main(){
8     printf("%d", fatorial(10));
9     return 0;
10 }
```

## 3. Código em Assembly feito pelo ChatGPT

```
1     .data
2 output_msg: .asciiz "O_fatorial_de_10_e:_"
3 newline:    .asciiz "\n"
4
5     .text
6     .globl main
7
```

```

8 main:
9     la $a0, output_msg
10    li $v0, 4
11    syscall
12
13    li $a0, 10
14    jal fatorial
15    move $a0, $v0
16
17    li $v0, 1
18    syscall
19
20    li $v0, 4
21    la $a0, newline
22    syscall
23
24    li $v0, 10
25    syscall
26
27 fatorial:
28     subu $sp, $sp, 8
29     sw $ra, 4($sp)
30     sw $a0, 0($sp)
31
32     li $v0, 1
33     lw $t0, 0($sp)
34     li $t1, 1
35
36 loop:
37     beq $t1, $t0, done
38     addi $t1, $t1, 1
39     mul $v0, $v0, $t1
40     j loop
41
42 done:
43     lw $ra, 4($sp)
44     addu $sp, $sp, 8
45     jr $ra

```

#### 4. Código em Assembly feito pelo Humano

```

1 .data
2 mens_r: .asciiz "resultado_=_\n"
3
4 .text
5 .globl main
6
7 main:
8     li $t0, 1
9     li $s1, 1

```

```

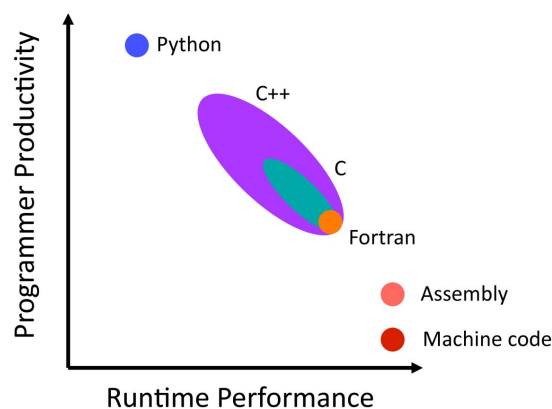
10
11     j fatorial
12
13 fatorial:
14
15     beq $t0, 11, fim
16     mult $s1, $t0
17     mflo $s1
18
19     addi $t0, $t0, 1
20     j fatorial
21
22 fim:
23
24     la $a0, mens_r
25     li $v0, 4
26     syscall
27
28     li $v0, 1
29     move $a0, $s1
30     syscall
31
32     li $v0, 10
33     syscall

```

## 5. Análise

### 5.1. Assembly x C

Ao comparar os códigos em Assembly com suas contrapartes em C, é evidente a discrepância de tamanho, pois o código C é traduzido para Assembly. Além disso, a proximidade do Assembly com a linguagem de máquina promove uma execução mais rápida e eficiente, já que se trata de uma linguagem de nível mais baixo, diretamente compreendida pela máquina.



**Figure 1. Comparação de desempenho e produtividade entre linguagens**

## 5.2. Assembly-GPT x Assembly-Humano

Entre o código feito pelo humano e o criado pelo ChatGPT, é nítida a diferença. Mesmo apesar de a IA necessitar de polimento e várias correções humanas durante a criação de códigos, ela conseguiu gerar um algoritmo mais complexo.

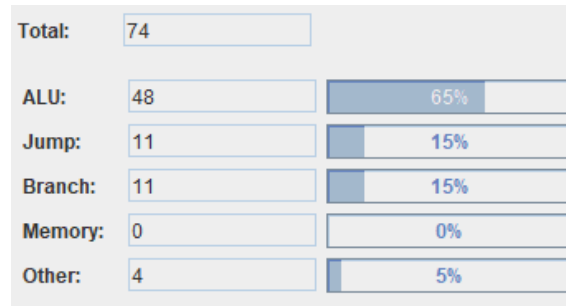


Figure 2. Ações do código feito pelo humano

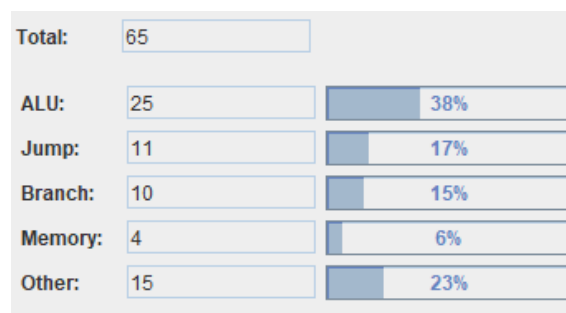


Figure 3. Ações do código feito pelo ChatGPT

Analisando ambas as execuções, é possível determinar, pelo número de ações, que a inteligência artificial, apesar de utilizar mais memória, consegue compilar o algoritmo mais rapidamente.

## 6. Conclusão

Após analisar os algoritmos em assembly, fica claro que, embora a máquina seja mais rápida, ela consome mais recursos computacionais. Além disso, o código feito pelo humano é mais compreensível para o público em geral. Essa constatação ressalta a importância não apenas da eficiência computacional, mas também da legibilidade e compreensão do código, especialmente quando se trata de alcançar uma audiência mais ampla.

## References

- [1] OpenAI. ChatGPT: A large-scale transformer-based language model. <https://openai.com/chatgpt>, 2021.
- [2] Automatic Fortran to C++ conversion with FABLE — RalfWGrosseKunstleve, ThomasC-Terwilliger, NicholasKSauter e PaulDAdams