

Tradução de código C para Assembly e comparações com o ChatGPT

Rafael de Souza Felisberto

Ciência da Computação - PUC-MG, Poços de Caldas, Brasil

Resumo: Este documento irá primeiro traduzir um código de cálculo de um número fatorial em linguagem C para a linguagem Assembly na arquitetura MIPS, após isso iremos analisar e comparar as diferenças entre o código feito pelo ser humano e do código feito pelo ChatGPT.

1. Introdução

Com a evolução das inteligências artificiais está ficando cada vez mais comum as comparações de códigos feitos por uma inteligência artificial e pelo ser humano, neste texto em específico iremos comentar sobre o ChatGPT^[1] e quais as diferenças de um código de cálculo de um número fatorial na linguagem Assembly em MIPS feito pelo chatbot e pelo ser humano, porém antes de tudo é necessário fazer a tradução do código a seguir em C.

2. Código de cálculo de um número fatorial em C

```
#include<stdio.h>
int fatorial(int n){
    if(n <= 1) return 1;
    return n * fatorial(n-1);
}

int main(){
    printf("%d", fatorial(10));
    return 0;
}
```

3. Código traduzido para Assembly feito pelo ser humano.

.data

```

.text
.globl main

main:
    li $t0, 1
    li $s1, 1

    j factorial

factorial:
    beq $t0, 11, end
    mult $s1, $t0
    mflo $s1

    addi $t0, $t0, 1
    j factorial

end:

    li $v0, 1
    move $a0, $s1
    syscall

    li $v0, 10
    Syscall

```

4. Código em Assembly feito pelo ChatGPT

```

.data
result: .asciiz "O fatorial de 10 é: "

```

```
newline: .ascii "\n"
```

```
.text
```

```
.globl main
```

```
main:
```

```
    # Inicialização de variáveis
```

```
    li $t0, 10      # Número cujo fatorial será calculado (10)
```

```
    li $t1, 1       # Variável para armazenar o fatorial, começa em 1
```

```
    li $t2, 1       # Contador para o loop
```

```
calc_fatorial:
```

```
    # Multiplica o fatorial atual pelo contador
```

```
    mul $t1, $t1, $t2
```

```
    # Incrementa o contador
```

```
    addi $t2, $t2, 1
```

```
    # Verifica se o contador alcançou o número
```

```
    ble $t2, $t0, calc_fatorial
```

```
    # Impressão do resultado
```

```
    li $v0, 4       # Código do syscall para imprimir string
```

```
    la $a0, result  # Endereço da string a ser impressa
```

```
    syscall
```

```
    li $v0, 1       # Código do syscall para imprimir inteiro
```

```
    move $a0, $t1   # Move o fatorial para o registrador de argumento
```

```
    syscall
```

```
# Impressão de uma nova linha  
  
li $v0, 4      # Código do syscall para imprimir string  
la $a0, newline # Endereço da string a ser impressa  
syscall  
  
  
# Encerramento do programa  
  
li $v0, 10     # Código do syscall para sair  
Syscall
```

5. Comparações entre os códigos:

5.1 Linguagem C x Assembly

Realizada a tradução do código de C para Assembly é notória a diferença entre estes, primeiramente vemos a maior complexidade da linguagem em Assembly, pois se trata de uma linguagem próxima a de máquina, o que interage diretamente com o hardware, o que é vantajoso quando se trata da velocidade de execução das tarefas, porém ao custo da dificuldade que o programador terá ao trabalhar com esta linguagem.

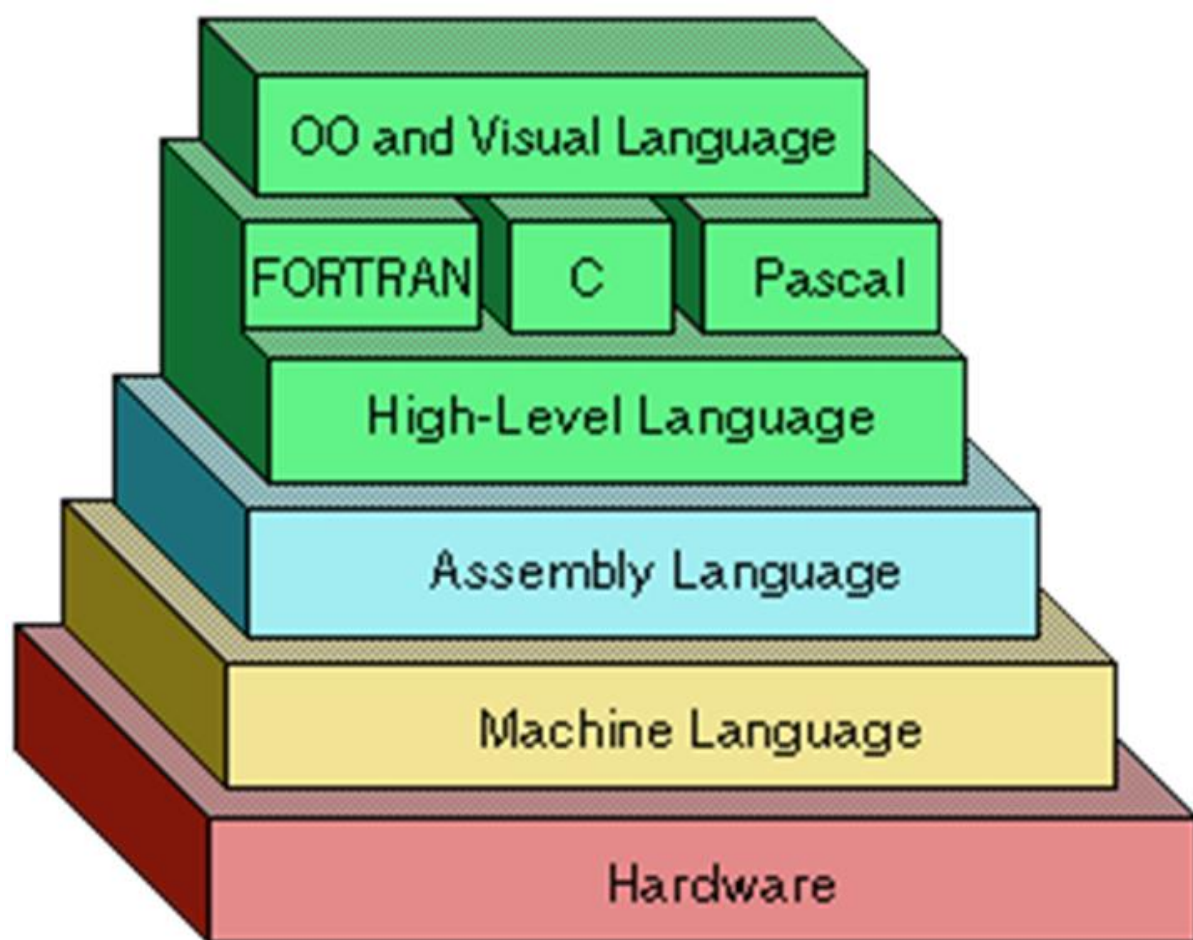
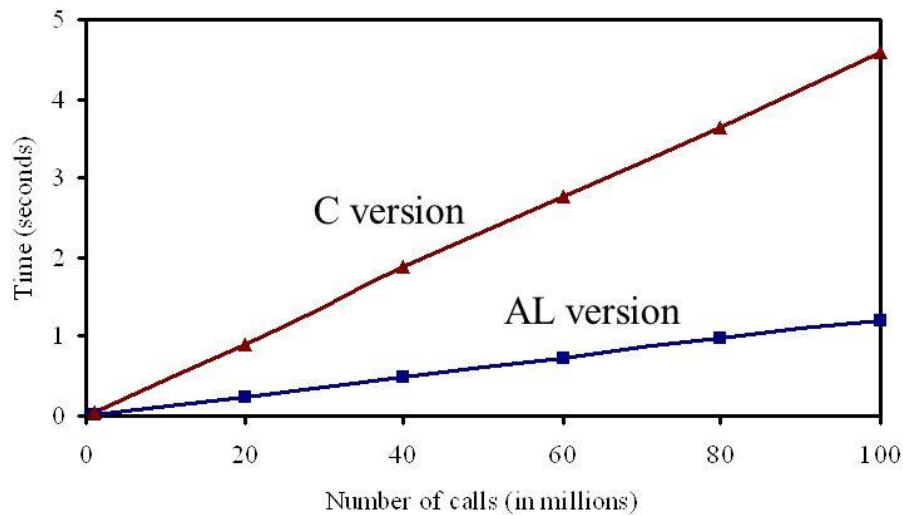


Figure 1[2]. Demonstração dos níveis de linguagens.

Performance: C versus Assembly Language



2005

To be used with S. Dandamudi,

"Introduction to Assembly

Language Programming,"

© 2005 S. Dandamudi

S. Dandamudi

Chapter 1: Page 16

Figure 2[3]. Comparação da velocidade de C e Assembly para cada chamada realizada.

5.2 Assembly realizado pelo ChatGPT x Assembly realizado pelo ser humano.

Ao ser pedido para a IA construir o código com base no outro, foram necessárias várias correções, porém é nítida a diferença na construção e na complexidade dos códigos, sendo o ChatGPT levando vantagem neste quesito.

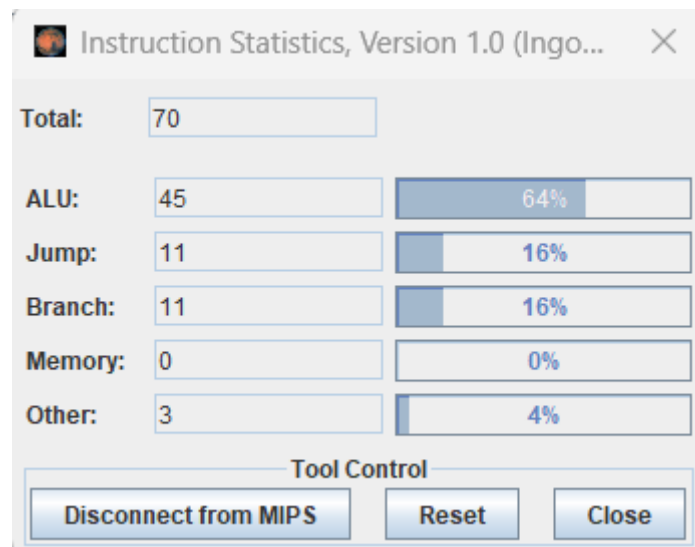


Figure 3. Código feito pelo Humano

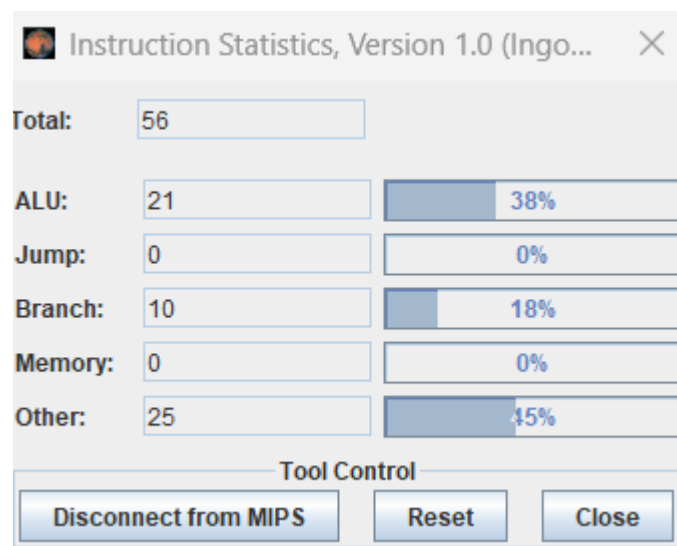


Figure 4. Código feito pelo ChatGPT

O ChatGPT consegue realizar um código bem mais complexo e necessita de bem menos ações para chegar no resultado necessário.

6. Conclusão

Podemos analisar com este artigo que os códigos feitos com a linguagem Assembly levam uma vantagem de velocidade computacional em relação aos códigos realizados em C. Agora ao analisarmos as diferenças entre os códigos do humano e da IA, fica claro que apesar de ser necessárias várias correções, o código fica mais complexo e mais bem aproveitado em relação ao ser humano.

7. Referências

[1] OpenAI. (2024). ChatGPT (May 16 version) [Large language model].
<https://chat.openai.com/chat> .

[2] <https://seattlewebsitedevelopers.medium.com/assembly-language-38e4b0edca0d>

[3] <https://slideplayer.com/slide/8144439/>