Diferença do Código Mips Manual x Chat GPT

Matheus Militão Santos

Pontifícia Universidade Católica de Minas Gerais - Poços de Caldas – MG – BRASIL

militaomatheus@outlook.com.br

**Resumo.** Este relatório compara a implementação de um código do cálculo fatorial feito por um humano e outro feito pelo chatGPT. A análise aborda aspectos como entrada e saída de dados, implementação da função fatorial, e a clareza da sintaxe e dos comentários.

**Abstract.** This report compares the implementation of a factorial calculation code written by a human and another generated by ChatGPT. The analysis addresses aspects such as input and output handling, implementation of the factorial function, and the clarity of syntax and comments.

# 1. Os códigos

**1.1 Código feito manualmente**

.data

mensagemI: .asciiz "Qual numero voce deseja calcular o fatorial ?"

mensagemR: .ascii "\n O fatorial e"

numero: .word 0

resp: .word 0

.text

.globl main

main:

# Ler o número do usuário

li $v0, 4

la $a0, mensagemI

syscall

li $v0, 5

syscall

sw $v0, numero

# Chamada da função fatorial

lw $a0, numero

jal fatorial

sw $v0, resp

# Mostrar o resultado

li $v0, 4

la $a0, mensagemR

syscall

li $v0, 1

lw $a0, resp

syscall

# Fim do programa

li $v0, 10

syscall

# Função fatorial

.globl fatorial

fatorial:

subu $sp, $sp, 8

sw $ra, ($sp)

sw $s0, 4($sp)

# Caso base

li $v0, 1

beq $a0, 0, saiFatorial

# Lógica (numero-1)

move $s0, $a0

sub $a0, $s0, 1

jal fatorial

mul $v0, $s0, $v0

saiFatorial:

lw $ra, ($sp)

lw $s0, 4($sp)

addu $sp, $sp, 8

jr $ra

**1.2 Código feito pelo ChatGPT**

.data

prompt: .asciiz "%d\n"

.text

.globl main

# Função fatorial

fatorial:

addi $sp, $sp, -8 # Cria espaço no stack para variáveis locais

sw $ra, 4($sp) # Salva o registrador de retorno

sw $a0, 0($sp) # Salva o argumento n

li $t0, 1 # Carrega 1 em $t0

ble $a0, $t0, base\_case # Se n <= 1, vai para base\_case

addi $a0, $a0, -1 # n = n - 1

jal fatorial # Chama fatorial(n-1)

lw $a0, 0($sp) # Restaura o valor de n

mul $v0, $a0, $v0 # Multiplica n \* fatorial(n-1)

j end\_recursion # Vai para end\_recursion

base\_case:

li $v0, 1 # Retorna 1

end\_recursion:

lw $ra, 4($sp) # Restaura o registrador de retorno

addi $sp, $sp, 8 # Libera espaço no stack

jr $ra # Retorna

# Função main

main:

addi $sp, $sp, -8 # Cria espaço no stack para variáveis locais

sw $ra, 4($sp) # Salva o registrador de retorno

li $a0, 10 # Carrega 10 em $a0

jal fatorial # Chama fatorial(10)

move $a0, $v0 # Move o resultado para $a0

la $a1, prompt # Carrega o endereço de prompt em $a1

li $v0, 4 # Código do serviço para imprimir string

syscall # Chama o serviço de sistema

lw $ra, 4($sp) # Restaura o registrador de retorno

addi $sp, $sp, 8 # Libera espaço no stack

li $v0, 10 # Código do serviço para encerrar o programa

syscall # Chama o serviço de sistema

# 2. Estrutura e organização

# 2.1 **Entrada e Saída de Dados:**

* 1. **Código 1:** Utiliza mensagens para interagir com o usuário, pedindo um número e exibindo o resultado. Usa syscalls para leitura e escrita.
  2. **Código 2:** Define um prompt para exibição do resultado, e o valor de entrada é fixo (10).

**2.2 Função Fatorial:**

* 1. Ambos implementam a recursão para cálculo do fatorial. O Código 1 usa **sub** para decrementar o valor e **beq** para o caso base. O Código 2 usa **addi** para decremento e **ble** para o caso base, o que inclui a verificação de **n <= 1** diretamente.

**2.3 Caso Base:**

* + **Código 1:** Define o caso base como **n == 0** retornando 1.
  + **Código 2:** Define o caso base como **n <= 1** retornando 1.

**2.4 Lógica de Recursão:**

* 1. **Código 1:** Realiza a chamada recursiva decrementando o valor e multiplicando o resultado na volta.
  2. **Código 2:** Semelhante, mas realiza a multiplicação e a chamada recursiva com uma abordagem mais clara utilizando etiquetas (**base\_case** e **end\_recursion**).

#### 3. **Sintaxe e Comentários:**

* **Código 1:** Inclui comentários explicativos para cada bloco de operação.
* **Código 2:** Utiliza uma estrutura clara de comentários, dividindo explicitamente a função fatorial e main, além de utilizar instruções mais claras (**addi**).

# 

# 4. Conclusão

Ambos os códigos implementam a função de cálculo do fatorial de maneira correta, utilizando técnicas de recursão. O Código 2, gerado pelo modelo ChatGPT, apresenta uma organização e clareza superiores, com uma estrutura de casos base e recursão bem definidas, além de um uso consistente das instruções de MIPS. O Código 1, apesar de funcional, possui uma abordagem mais manual na manipulação de pilha e lógica de decremento.

# Referencias

* Organização do texto conforme norma da SBC
* Conhecimento aplicado para tais informações adquirido em aula com o professor Harrison Herman Silva e atraves da bibliográfia [Patterson, D. A., & Hennessy, J. L. Organizacao e Projeto de Computadores. Elsevier]