

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO  
ESPÍRITO SANTO - IFES

João Pedro Garcia Pereira

Paulo Ricardo Pereira Gomes

Rodrigo Couto Rodrigues

**Relatório do Trabalho de Assembly/C**

Serra

2021

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO  
ESPÍRITO SANTO - IFES

João Pedro Garcia Pereira

Paulo Ricardo Pereira Gomes

Rodrigo Couto Rodrigues

## **Relatório do Trabalho de Assembly/C**

**Relatório do aprendizado em assembly proposto  
por esse trabalho e comparações com a  
linguagem C**

**Professor: Flávio Giralde**

Serra 2021

## **Introdução**

Assembly é uma linguagem de baixo nível, já a linguagem C é de alto nível, este trabalho tem como objetivo, entender como funciona a linguagem assembly, suas peculiaridades e conceitos, utilizando os conhecimentos obtidos em arquitetura de computadores e comparar com a linguagem C alguns aspectos.

# Relatório

## **Alguns exemplos de você sequer precisa saber que aquilo existe em Assembly ao implementar em C:**

Um deles é o fato de Assembly ser uma linguagem de baixo nível, isso significa que você para programar em Assembly é necessário ter um conhecimento sobre a estrutura do processador no qual você irá utilizar para programar, neste trabalho utilizamos o microprocessador 8086, com base no conhecimento adquirido em arquitetura de computadores, é possível notar o progresso da tecnologia, suas particularidades, suas próprias características em geral e códigos.No caso do Assembly:

### **Registradores**

Registradores de dados

AX, BX, CX e DX

Registradores ponteiros

BP,IP,SP,SI e DI

Registradores de segmento

CS,DS,SS e ES

### **Acesso a memória**

### **Interrupções**

## **Comparação da quantidade de instruções relevantes presentes em C/Assembly:**

A instrução “print” existente no C, facilita mostrar os componentes do programa na tela,o “scanf” que possibilita o input no programa.

Assembly possui as instruções mnemônicos que facilita a codificação, alguns exemplos são ADD,SUB,MOV,etc

## **Explicação do funcionamento básico de todos os macros/procedimentos que foram copiados do inc\emu8086.inc**

Utilizamos o PUTC MACRO este macro tem a função de imprimir um caractere em AL e avançar a posição atual do cursor.

Utilizamos também o SCAN\_NUM PROC esse procedimento permite que o usuário digite vários dígitos do teclado, e armazena o resultado no registro CX.

E também PRINT\_NUM\_UNPROC esse procedimento tem a função de imprimir um número em AX porém ele imprime valores de 0 até 65535 e não apenas 1 caractere.

## **Avaliação do grupo do tempo necessário para cada implementação. Tome a implementação em C como referência e estime o percentual de tempo a mais necessário para a implementação em Assembly.**

Como temos uma matéria basicamente focada em C, implementar o que queríamos nessa linguagem foi muito mais rápido e fácil, em Assembly mesmo com os vídeos e os exemplos disponíveis tivemos dificuldade para assimilar e por isso demoramos mais que o dobro do tempo para implementar em assembly.

## **Conclusão**

Em suma, programar em assembly, necessita um conhecimento mais aprofundado da arquitetura do computador, processador e etc, sendo mais difícil de programar, códigos mais extensos, porém na maioria das vezes mais eficientes. Por mais que o código não seja tão intuitivo, a medida que você vai praticando, dá para entender alguns conceitos e avança na linguagem, comparado a linguagem C, é necessário um gasto maior de tempo e esforço para conseguir entender.