**Engenharia da Computação**

**Programação de Hardware**

**2º Semestre 2021**

**Implementação de Assembler em linguagem C**

**Ailton Fernandes do Rosário Junior – RA 171912**

1. ***Como Implementar Assembler em linguagem C* ?**

Caso esteja necessitando de automatizar sua programação em assembler utilizando C, e tenha dúvida de como fazer a implementação, nesse texto explicaremos de forma fácil como implementar assembler em C.

Após criar o projeto em “**GCC C Executable Project**”, adicionaremos os arquivos de assembler, que devem estar com a extensão “***.S***”.

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Texto

Descrição gerada automaticamenteTexto

Descrição gerada automaticamenteUtilizaremos como exemplo um código em assembler para fazer um LED piscar com delay de aproximadamente 1s entre ligado e desligado.

Caso venha a ter problemas de “***out of range***”, utilize um registrador para guardar o valor binário assim como feito com o r17 e r18 na figura acima, exemplo: ***r18 = 0b11110000***, em nosso caso resolveu esse problema.

Tendo feito isso, a seguir começaremos a implementar o Assembler à linguagem em C.

Primeiramente incluiremos a biblioteca avr/io.h em todos os arquivos, para incluir utilizamos o comando ***#include <avr/io.h>.***

Após incluir a biblioteca, definiremos as diretrizes globais, no caso habilitaremos a função em assembler para ser utilizada em qualquer momento, para fazer isso utilizamos o comando .global seguido do nome da função a ser utilizada, no exemplo utilizamos o comando *“****.global INICIO***”.

No terceiro passo devemos permitir o acesso da função externa para que possa ser utilizada no arquivo principal em C, para isso utilizamos o comando **“*extern*”**, que permite o acesso da função externa seguido de **“*void*”** em nosso caso pois não haverá retorno, mas poderia ser char caso tenha retorno de valor e por último a função a ser chamada, esse comando deve ser feito fora das funções, preferencialmente na parte global do código em C para que possa ser utilizado em qualquer parte do código. Exemplo do comando: ***extern void INICIO();***

Caso precise, pode ser utilizado o parâmetro char entre o parêntese na função, esse comando deve ser feito na parte global código.

E para finalizar, escrevemos dentro da função principal **int main()**, a função a ser chamada, a ***INICIO();***.

Texto

Descrição gerada automaticamente

1. **Mas porque é preciso esse tipo de implementação ?**

Utilizamos esse tipo de implementação, quando queremos economizar comandos por sua vez economizando espaço de memória, a linguagem em C carrega muitos comando que não são utilizados dentro da aplicação desenvolvida esses comando acabam por gerar um gasto de memória que são evitados ao utilizar a programação em assembler já que é apenas executado os comando necessários para a aplicação, em programas pequenos não há a necessidade da utilização do código em assembler, porém em programas mais extensos pode ser necessário a utilização pois dependendo do tamanho do programa e do espaço de memória do microcontrolador utilizado faz-se necessário a economia de espaço.

1. **Segue em anexo o link para o repositório do GitHub para a consulta do código desenvolvido:**

https://github.com/Ailtondiz/Programacao\_de\_Hardware.git