**Noções de Assembly**

Assembly é uma linguagem de programação de baixo nível, dá acesso a todos os detalhes de funcionamento de um processador, por este fato esta linguagem não é portável, cada programa desenvolvido é criado para uma arquitetura em específica, isso significa que o programa desenvolvido para uma arquitetura não será funcional em outra arquitetura, tornando a programação mais complexa, mais ainda esta linguagem de programação é um pouco confusa, deixando sua manutenção uma tarefa árdua, porém por ser uma linguagem de baixo nível garante que os programas serão leves e tirarão o máximo do proveito, a melhor performance possível.

O Assembly é muito utilizado em microcontroladores, pois sua capacidade de armazenamento é mínima e necessita que o programa seja o mais leve possível e tire o máximo aproveitamento.

**Como Compilar um programa desenvolvido em Assembly?**

Assim como todas as linguagens de programação, há um compilador (no caso do Assembly o compilador é chamado de Assembler) que interpreta as instruções para a linguagem de máquina. Existe o Compilador NASM (Netwide Assembler) que está disponível no site <http://www.nasm.us/>. Este compilador utiliza linhas de comandos via Prompt e está disponível para Linux e Windows totalmente gratuito e atualizado.

Neste artigo utilizaremos o Assembly integrado ao código-fonte do C/C++ usando o IDE DevC++, compilador TDM-GCC, segue abaixo um exemplo desta integração:

// Declarar Bibliotecas

#include <stdio.h>

#include <stdlib.h>

**int** soma (**int** a, **int** b, **int** c); // Declarar Função

// Atribuir instruções a função

**int** soma (**int** a, **int** b, **int** c){

**int** x; // variavel de retorno da função

// realiza a soma em assembly - adiciona os valores dos registradores ebx, ecx ao registrador eax somando os valores

//instrução em assembly

**asm volatile** (**"add %%ebx, %%eax; add %%ecx, %%eax;"**

:**"=a"**(x) //saida de dados do registrador eAx

:**"a"**(a), **"b"**(b),**"c"**(c) //entrada de dados aos registradores eAx, eBx, eCx

: **"memory"**

);

**retur**n x; //retorno do resultado

}

**int** main(**void**){

printf(**"%d"**,soma(1,2,10)); // retorna o valor 13 que é a soma de 1+2+10

system(**"pause>null"**);

}

**Referencias:**

Assembly Progressivo. Disponível em <http://www.assemblyprogressivo.net/>. Visualizado em 15 de setembro de 2016.

Registradores Assembly. Disponível em <http://assemblytutorial.wikidot.com/registers>. Visualizado em 17 de setembro de 2016.

What's wrong with this assemly code? – Ubuntu Forums. Disponível em <https://ubuntuforums.org/showthread.php?t=144568>. Visualizado em 15 de setembro de 2016.