Trabalho – linguagens de montagem

Resumo

Relatório da realização de atividade prática englobando os vários conceitos abordados durante o curso.  
Criar aplicações com interfaces entre linguagens de montagem e linguagens de alto nível  
(Assembly e Linguagem C).  
Comparar tempo de execução do código em C e Assembly, visando otimização do código  
Assembly.



Jonathan Henrique

Léo Eduardo

**PROGRAMA EM C**

As informações de linhas e colunas das matrizes estão no arquivo **tamanho.h**.   
As funções de manipulação da matriz estão no arquivo **matriz.h**.

A multiplicação de matrizes só pode ocorrer se o número de colunas da primeira for igual ao número de linhas da outra matriz.

Satisfeita essa comparação, preenchemos as matrizes.  
Multiplicamos a primeira matriz por um valor inteiro, que no caso é 5.   
O resultado será o menor valor da diagonal principal das matrizes multiplicadas.

***preencher\_matriz*** recebe a primeira, a segunda matriz e seed.  
Esta função percorrerá as linhas e colunas das matrizes e a preencherá com valores arbitrários utilizando ***rand()****.*

***multiplica\_por\_inteiro*** recebe a primeira matriz, a qual será multiplicada pelo inteiro 5.  
O laço garante que todas as colunas de todas as linhas serão multiplicadas e a posição da matriz atualizada com este novo valor.

Para exibir o resultado que é o menor valor da diagonal principal, chamamos a função ***menor\_valor\_diagonal\_principal\_mutiplicada***.

***menor\_valor\_diagonal\_principal\_mutiplicada*** recebe a primeira e a segunda matriz.  
**tamanho\_diagonal** receberá a quantidade de linhas da primeira matriz e caso a quantidade de colunas da segunda seja maior, **tamanho\_diagonal** é atualizado.  
A variável **c** carregará o item da diagonal e dentro do laço multiplicaremos as linhas pelas colunas com a função ***multiplica\_linha\_por\_coluna*.** Caso o retorno salvo em **aux** seja menor que **resultado**, atualizamos esta última garantindo que sempre contenha o menor valor da diagonal principal.

***multiplica\_linha\_por\_coluna*** recebe a primeira matriz, a segunda matriz, o item da diagonal e o tamanho da diagonal.  
Dentro do laço **resultado** recebe o retorno da multiplicação das linhas por colunas.

**PROGRAMA EM ASSEMBLY**

No arquivo **matriz.asm** temos as mesmas funções contida em **matriz.h**.

A multiplicação de matrizes só pode ocorrer se o número de colunas da primeira for igual ao número de linhas da outra matriz.

Satisfeita essa comparação, preenchemos as matrizes.  
Multiplicamos a primeira matriz por um valor inteiro, que no caso é 5.   
O resultado será o menor valor da diagonal principal das matrizes multiplicadas.

No NASM, para multiplicarmos pelo inteiro 5 recorremos à função ***nasm\_multiplica\_por\_inteiro***.

***nasm\_multiplica\_por\_inteiro:***

**X = +8number = +12  
A\_linhas = +16  
A\_colunas = +20  
l = -4  
c = -8**

Antes de tudo, inicializamos **l**=0.

***\_for\_linhas\_mul:***Comparamos **l** com o tamanho das linhas da primeira matriz. Caso seja maior, significa que terminamos. Caso contrário, pula para a função ***\_body\_for\_linhas\_mul*** onde inicializamos **c**=0 e pulamos para a função ***\_for\_colunas\_mul****.*

***\_for\_colunas\_mul:***Comparamos **c** com o tamanho das colunas da primeira matriz. Caso seja maior, significa que percorremos todas as colunas, incrementamos **l** e o recursão se repete em ***\_for\_linhas\_mul***.   
Caso **c** seja menor, recorremos à função ***\_body\_for\_colunas\_mul*** que é responsável por multiplicar as posições pelo inteiro.

***\_body\_for\_colunas\_mul:***Percorremos todas as colunas da matriz e quando **c** for igual ao tamanho das colunas, incrementamos **l** e retomamos o laço. Isto garante que todas as posições serão multiplicadas por 5 e salvas na matriz. Por fim, retornamos à ***\_for\_linhas\_mul*** para comparação de **l** com o tamanho das linhas da primeira matriz.

Após multiplicada pelo inteiro, recorremos à função ***nasm\_menor\_valor\_diagonal\_principal\_mutiplicada*** onde buscaremos o menor valor da diagonal.

***nasm\_menor\_valor\_diagonal\_principal\_mutiplicada:***

**X = +8  
Y = +12  
A\_linhas = +16  
A\_colunas = +20  
C\_linhas = +24  
C\_colunas = +28  
resultado = -4  
c = -8  
tamanho\_diagonal = -12  
  
aux não necessário, pois EAX já armazena o resultado**

O tamanho da diagonal inicialmente é o tamanho das linhas da primeira matriz. Caso o tamanho da diagonal for menor que o tamanho das colunas da segunda matriz, atualiza-se o valor. O **resultado** inicialmente recebe -1 e **c**=0. Feitas inicializações pulamos para ***\_for\_item\_diagonal***.

***\_for\_item\_diagonal:***Verificamos se **c** é menor que o tamanho da diagonal, caso seja maior significa que percorremos a diagonal e terminamos. Caso seja menor pulamos para ***\_body\_for\_item\_diagonal*** que é encarregada de fazer a *call* da função ***nasm\_multiplica\_linha\_por\_coluna,*** a qual multiplicará as linhas pelas colunas. Após isso, salva o retorno, contido em **aux**, em **resultado** caso seja menor que o valor atual da diagonal*.* Neste ponto, incrementa-se **c** e o laço se repete em***\_for\_item\_diagonal*** .

***nasm\_multiplica\_linha\_por\_coluna:***

**X = +8  
Y = +12  
pos = +16  
max = +20  
A\_linhas = +24  
A\_colunas = +28  
C\_linhas = +32  
C\_colunas = +36  
resultado = -4  
c = -8**

Inicializamos **resultado**=0 e **c**=0 e pulamos para ***\_for\_mul\_linha***.

***\_for\_mul\_linha:***Comparamos **c** com o tamanho da diagonal, caso seja menor pulamos para ***\_body\_for\_mul\_linha****.* Caso seja maior, significa que percorremos a diagonal e retornamos o resultado.

***\_body\_for\_mul\_linha:***Comparamos **c** com o tamanho das colunas da primeira matriz e caso seja maior, incrementamos **c** e retomamos o laço em ***\_for\_mul\_linha***. Caso seja menor, comparamos **c** com o tamanho das linhas da segunda matriz e caso seja maior, retomamos o laço em ***\_for\_mul\_linha***. Caso seja menor, multiplicamos as posições das matrizes e incrementamos **c**.  
Quando **c** for maior que o tamanho da diagonal, percorremos todas as posições e retornamos o resultado para **aux**.

**RESULTADO DOS TESTES**

Abaixo a tabela com os resultados em msec, exceto o tamanho da matriz.

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho das Matrizes | Código C | Código NASM | Código GAS |
| (5,5) | 0,002 | 0,003 | 0,003 |
| (15, 15) | 0,004 | 0,007 | 0,007 |
| (30, 30) | 0,024 | 0,035 | 0,028 |
| (60, 60) | 0,074 | 0,087 | 0,085 |
| (120, 120) | 0,338 | 0,366 | 0,351 |
| (240, 240) | 1,552 | 1,455 | 1,121 |
| (500, 500) | 2,219 | 2,601 | 2,151 |
| (650, 650) | 3,889 | 4,136 | 4,262 |
| (750, 750) | 5,558 | 6,244 | 6,312 |
| (900, 900) | 7,933 | 10,329 | 9,879 |
| (1000, 1000) | 8,308 | 12,003 | 10,546 |