

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO

TRABALHO DE PROGRAMAÇÃO EM ASSEMBLY PARA ARQUITETURA MIPS32

MARCUS PAULO SOARES DANTAS

NATAL-RN
2018

MARCUS PAULO SOARES DANTAS

TRABALHO DE PROGRAMAÇÃO EM ASSEMBLY PARA ARQUITETURA MIPS32

NATAL-RN

2018

SUMÁRIO

1	INTRODUÇÃO	3
2	DESCRIÇÃO DO CÓDIGO	4
3	CONCLUSÃO	6
	REFERÊNCIAS	7

1 INTRODUÇÃO

O objetivo dessa atividade é a aplicação de conceitos aprendidos em aula em problema práticos que se assemelham a situações reais. Diante disso nos foi proposto o desenvolvimento de um histograma em assembly, a partir de um pseudo código fornecido tivemos que desenvolver um código capaz de processar o histograma de uma imagem fornecida previamente.

2 DESCRIÇÃO DO CÓDIGO

Levando em consideração o código com algumas declarações de variáveis disponibilizado pelo professor abaixo no Listing 2.1 encontra-se apenas o escopo do main. Em relação as declarações de variáveis foi criada apenas uma nova variável H que armazena os 256 possíveis valores do histograma, ela ocupa 1024 bits onde cada inteiro ocupa o equivalente a 4 bits.

Listing 2.1 – Código fonte em Scilab

```
main:
    li $t0, 0 # CONTADOR PARA O VETOR H.
    lw $t1, vmaximo # CARREGA O VALOR DO VMAXIMO PARA MEMORIA. 255
    la $s1, H # Carrega o vetor H no registrador s1
    inicioForH: # Inicio do vetor para preencher H com zeros. for(int i=0; i<256; i++)
        ble $t0, $t1, forH # Branch se $t0 menor ou igual a $t1 va para instrucao forH
        j fimForH # Jump para fim do for caso $t1 = 256
    forH:
        sw $zero, 0($s1) # Adiciona zero a posicao atual do vetor.
        addi $s1, $s1, 4 # Move 4 bytes na posicao do vetor para acessar a nova posicao
        addi $t0, $t0, 1 # Incrementa mais um no for
        j inicioForH # Jump para inicio do for.
    fimForH:

    lw $t1, colunas # Carregando o tamanho das colunas do vetor.
    lw $t2, linhas # Carregando o tamanho das linhas do vetor.

    mul $t3, $t1, $t2 # fazendo multiplicacao das linhas pelas colunas para o segundo for

    la $s1, H # Carregando H zerado no reg s1
    la $s2, imagem # Carrega a imagem no registrador s2
    li $t0, 0 # CONTADOR PARA O VETOR H.
    li $t5, 0 # CONTADOR PARA O DESLOCADOR H.

    inicioForHistograma: # Inicio do for para execucao do histograma
        blt $t0, $t3, forHistograma # Branch se $t0 menor a $t3 va para instrucao forHistograma
        j fimForHistograma # Jump para fim do for caso $t1 = 256
    forHistograma:
        lw $t4, 0($s2) # Carregando valor da imagem no registrador t4
        inicioForAndarH:
            ble $t5, $t4, forAndarH # Branch se $t0 menor ou igual a $t1 va para instrucao
            j fimForAndarH
        forAndarH:
            addi $s1, $s1, 4 # Move 4 bytes na posicao do vetor para acessar a nova posicao
            addi $t5, $t5, 1 # Incrementa mais um no for
        fimForAndarH:
        lw $t6, 0($s1)
        addi $t7, $t6, 1
        sw $t7, 0($s1)
        addi $t0, $t0, 1 # Incrementa mais um no for
        addi $s2, $s2, 4 # Move 4 bytes na posicao do vetor para acessar a nova posicao
        la $s1, H # Carregando H zerado no reg s1
        la $t7, 0
        j inicioForHistograma # Jump para inicio do for.
    fimForHistograma:
```

No inicio do código foram feitas algumas operações para salvar os valores dos contador

(Contador iniciando com 0) e o valor máximo para o histograma assim como foi carregado no registrador s1 o endereço do vetor H. Após essas instruções foi criada uma instrução inicioForH que tem um condicional que permite a continuidade nessa instrução se t0 for menor ou igual a 255 quando o contador for 256 então ocorrerá um jump para instrução fimForH.

Após o fim da instrução do primeiro for carregamos os valores máximos de colunas e linhas, fazemos uma pequena multiplicação para saber o valor máximo que a imagem pode assumir. Depois carregamos o vetor H na memória assim com a imagem além de iniciar os contadores para os for's.

No incio do for histograma temos uma condição inicial para verificar se os contadores estão dentro da faixa desejada, se estiverem vamos para instrução forHistogram nela carregamos o valor que a imagem possui na posição atual do for e iniciamos uma nova instrução inicioForAndarH que serve para deslocar o vetor H até a posição equivalente ao pixel da imagem acessado, após essa instrução ser finalizada carregamos o valor que o vetor H possui em um registrador somamos mais um e salvamos na posição atual do vetor H, com esse processo realizado incrementamos o contador do for, movemos a imagem e voltamos a posição 0 do vetor H, o for irá se repetir até o histograma ser finalizado.

3 CONCLUSÃO

A ideia de programar em assembly é realmente desafiadora, levando em consideração que instruções triviais como um for necessitam de uma maior atenção para ser implementadas nessa linguagem o programador é desafiado a "pensar" um pouco mais no desenvolvimento no seu código, o código do histograma fornecido pelo professor poderia ser reproduzido em poucas linhas em uma linguagem de mais alto nível porém em assembly esse código se mostrou realmente desafiador sendo um ótimo trabalho avaliativo para primeira unidade da disciplina de Arquitetura de Computadores.

REFERÊNCIAS

QtSpim Tutorial. University of the Pacific. Disponível em: <<http://ecs-network.serv.pacific.edu/ecp170/tutorials/qtspim-tutorial>>

Programmed Introduction to MIPS Assembly Language. Central Connecticut State University. Disponível em :<<https://chortle.ccsu.edu/AssemblyTutorial/index.html>>