

T2B – Programação em Linguagem de Montagem MIPS

Descrição

Para desenvolvimento da solução foi utilizado como base o template disponibilizado pelo professor. O programa foi dividido em diversos labels que realizam chamadas de funções, desvios condicionais e laços de repetição para solução do problema descrito.

Durante o processo de codificação foi separado alguns registradores para uso exclusivo no acesso a dados na memória e chamadas de funções.

Registradores de acesso a Memória:

- \$t0 - usado para carregar endereço da matriz de ANOS
- \$t1 - usado para carregar endereço da variável LINHA
- \$t2 - usado para carregar endereço da variável COLUNA
- \$s0 - usado para carregar endereço da variável COPA_CNT
- \$s1 - usado para carregar endereço do vetor COPA_ANOS
- \$s2 - usado para carregar endereço da variável OLIM_CNT
- \$s3 - usado para carregar endereço do vetor OLIM_ANOS

Registradores exclusivos para funções:

- \$a0 - usado para carregar os dados a serem exibidos
- \$a1 - usado para setar o Syscall da função imprimir
- \$a2 - usado como parâmetro das funções f_copa e f_olimpiada
- \$v0 - usado para as chamadas de Syscall dentro das funções
- \$v1 - usado como retorno das funções f_copa e f_olimpiada

Código

O label *main* começa com a definição dos registradores que irão armazenar os endereços das variáveis, vetores e matriz utilizadas pelo programa de modo que estes registradores foi deixado para uso exclusivo. Já o label *linha* contém o trecho do código responsável por fazer a leitura da linha informada verificar se ela é válida através das instruções *blez* e *bgt* (testando se o valor informado encontra-se dentro do range de linhas). Caso passe pela verificação é calculado a partir de qual endereço (posição na matriz).

```
main:
    la    $t0, ANOS          # Carrega o endereço para "vetor" ANOS
    la    $t1, LINHA         # Carrega o endereço para LINHA
    la    $t2, COLUNA        # Carrega o endereço para COLUNA
    la    $s0, COPA_CNT      # Carrega o endereço para COPA_CNT
    la    $s1, COPA_ANOS     # Carrega o endereço para vetor COPA_ANOS
    la    $s2, OLIM_CNT      # Carrega o endereço para OLIM_CNT
    la    $s3, OLIM_ANOS     # Carrega o endereço para vetor OLIM_ANOS
    lw    $s6, 0($s0)         # Carrega a Quantidade de Copas
    lw    $s7, 0($s2)         # Carrega a Quantidade de Olimpíadas
    lw    $t1, 0($t1)         # Carrega o valor da LINHA
    lw    $t2, 0($t2)         # Carrega o valor da COLUNA

linha:
    la    $a0, TEXT0_1       # Carrega o texto a ser exibido
    li    $a1, 4              # Parametro Syscall
    jal   imprimir           # Chamada da funcao imprimir TEXT0_1
    li    $v0, 5              # Preparo programa para capturar valor
    syscall                  # Chamada do sistema para receber inteiro
    blez  $v0, linha         # if valor informado <= 0 salta para linha
    bgt   $v0, $t1, linha     # if valor informado >= LINHA salta para linha
    move  $t3, $v0            # Carrega o valor da linha desejada
    li    $t4, 4              # Distancia entre cada endereco de memoria
    mul   $t4, $t4, $t2       # Calcula quantos bits e acrescentado para o endereco de memoria da linha desejada
    addi  $t3, $t3, -1        # Subtrai 1 da linha informada para calcular o endereco primeiro ano
    mul   $t4, $t4, $t3       # Calcula o endereco do primeiro ano a ser manipulado
    add   $t0, $t0, $t4       # Recebe primeiro ano a ser calculado
```

O label *loop1* é a primeira da parte do laço que realiza a chamada das funções *f_copa* e *f_olimpiada* verificando se o ano que foi carregado a partir do endereço de memória citado acima e faz as manipulações testando se houve ou não ocorrência de copa ou olimpiada no ano informado. Vale ressaltar que como não existe casos em que tenha ocorrido copa do mundo e olimpíadas o programa assim que encontra uma ocorrência ele a adiciona e salta para o próximo ano a ser testado. Isto pode ser visto na divisão do código através dos labels *add_copa*, *add_olim* e *loop2*. Durante o processo de repetição há dois registradores responsáveis por armazenar as ocorrências de copa do mundo e olimpíadas. Também é separado um registrador responsável por verificar se a linha que esta sendo testada chegou ao final.

```
loop1:
    blez  $t2, final          # Verifica se o indice coluna e igual ou menor que zero
    lw    $a2, 0($t0)         # Carrega o ano em $a2 para chamada de funcoes
    jal   f_copa               # Chamada da funcao copa
    bne   $v1, $zero, add_copa # Verifica se a funcao retornou ano : Salta para adicionar ano
    jal   f_olimpiada         # Chamada da funcao olimpiada
    bne   $v1, $zero, add_olim # Verifica se a funcao retornou ano: Salta para adicionar ano
    j     loop2               # Caso nao haja ocorrencia salta para loop2

add_copa:
    # Adicionar ano na Copa
    addiu $s6, $s6, 1         # Adiciona uma ocorrencia no contador Copa
    sw    $v1, 0($s1)         # Armazena no vetor ANOS_COPA
    add   $s1, $s1, 4          # Aponta o endereco para proxima posicao do Vetor Copa
    j     loop2               # Continua loop

add_olim:
    # Adicionar ano na Olimpiada
    addiu $s7, $s7, 1         # Adiciona uma ocorrencia no contador Olimpíadas
    sw    $v1, 0($s3)         # Armazena no vetor ANOS_OLIMPIADAS
    add   $s3, $s3, 4          # Aponta o endereco para proxima posicao do Vetor Olimpíadas
    j     loop2               # Continua loop

loop2:
    add   $t0, $t0, 4          # Passa para a proxima posicao da linha
    addiu $t2, $t2, -1        # Decrementa o indice da coluna
    j     loop1               # Salta sempre para o loop
```

Logo após sair do laço de repetição o label *final* se encaminha para exibição do resultado obtido a partir dos calculos realizados anteriormente. Nota-se que apenas neste momento é salvo a quantidade nas quais ocorreram copa e olimpíadas e não no processo de repetição. Seguindo no código vemos a utilização da função *imprimir* para mostrar as ocorrências de copa do mundo na linha informada. O label *loop3* é responsável por percorrer as ocorrências de copas e exibi-las. Então avança para o label *final2* no qual é realizado as mesmas chamadas do *final* só que referente as ocorrências de olimpíadas. Também existe um label *loop4* que percorre todas ocorrências de olimpíadas para exibir. Ao finalizar o *loop4* o programa avança para o label *exit* que simplesmente encerra o programa.

Funções

Para desenvolvimento das funções foi utilizado por padrão a passagem dos parametros através dos registradores \$a0 - \$a3 e retorno nos \$v0 e \$v1. Abaixo vamos ver como foi descrito as funções *imprimir*, *f_copa* e *f_olimpiada*.

A função *imprimir* tem como base receber em \$a0 a mensagem a ser exibida independente do tipo (string, int, float entre outros) e em \$a1 recebe o Syscall correspondente ao tipo de dado a ser exibido.

```
#####  
# Recebe a mensagem em $a0 e o tipo de Syscall em $a1  
imprimir:  
    move    $v0, $a1          # Carrega o Syscall passado pelo $a1  
    syscall  
    jr      $ra
```

As funções *f_copa* e *f_olimpiada* tem como base receber o ano a ser testado no registrador \$a2 e retorna através do registrado \$v1 se houve ou não ocorrência. Durante a manipulação da função que verifica ocorrências de olimpíadas as exceções de anos foram tratadas antes de realizar o cálculo. Assim é somente testado os anos que se encontram no intervalo pré definido na especificação.

```
#####  
# Recebe o ano a ser verificado em $a2 e retorna o ano em $v1 caso haja ocorrência  
f_copa:  
    li      $s4, 1930          # Carrega o inicio do intervalo  
    blt     $a2, $s4, copa_false # Verifica se o ano e menor que 1930 : Salta para nao armazenar  
    li      $s4, 2018          # Carrega o final do intervalo  
    bgt     $a2, $s4, copa_false # Verifica se o ano e maior que 2018 : Salta para nao armazenar  
    subu    $s4, $s4, $a2       # 2018 - Ano  
copa_loop:  
    beq     $s4, $zero, copa_true # Se resto for igual a 0 passa  
    subu    $s4, $s4, 4          # Subtrai 4  
    bltz    $s4, copa_false     # Se resto for menor que 0 nao passa  
    j       copa_loop          # Continua testando  
copa_true:  
    move    $v1, $a2           # Armazena o ano em $v1  
    j       copa_fim           # Segue para fim da funcao  
copa_false:  
    move    $v1, $zero         # Armazena zero em $v1  
    j       copa_fim           # Segue para fim da funcao  
copa_fim:  
    jr      $ra
```

Conforme podemos ver acima a função foi separada em três etapas uma responsável por realizar os calculos, uma parte responsável por setar o retorno da função e outra para retornar ao programa principal.