

Lista de exercícios 2

8

Considere o seguinte código em MIPS assembly:

```
add $s0, $s0, $s1
add $s0, $s0, $s2
add $s0, $s0, $s3
add $s0, $s0, $s4
sw  $s0, 12($s7)
```

Assuma que os registradores `$s0`, `$s1`, `$s2`, `$s3` e `$s4` estão associados às variáveis `f`, `g`, `h`, `i` e `j`. Assuma que o registrador `$s7` contém o endereço de um array designado por `A`. Determine o código correspondente em linguagem C com o mínimo de linhas de código possível.

Resposta 8

Inicialmente é importante saber o que acontece em cada linha de código MIPS Assembly, logo, vamos comentar e entender o que está acontecendo:

```
add $s0, $s0, $s1 # Adiciona $s0 com $s1 e armazena em $s0 (s0 = s0 + s1)
add $s0, $s0, $s2 # Adiciona $s0 com $s2 e armazena em $s0 (s0 = s0 + s2)
add $s0, $s0, $s3 # Adiciona $s0 com $s3 e armazena em $s0 (s0 = s0 + s3)
add $s0, $s0, $s4 # Adiciona $s0 com $s4 e armazena em $s0 (s0 = s0 + s4)
sw  $s0, 12($s7) # Armazena o valor de $s0 no Array de endereço armazenado
em $s7 na posição 3 (4 bytes por palavra)
```

Dessa forma, podemos observar que as 4 primeiras linhas representam uma estrutura de soma, e que estamos somando os valores dos registradores `$s0`, `$s1`, `$s2`, `$s3` e `s4`, acumulando a soma em `$s0`. Em seguida, na instrução `sw`, desejamos armazenar o que acumulamos da operação em nosso Array **A**, e que deve ser inserido na posição 3 (considerando **A** um Array de palavras), para isso observe que deve ser inserido a partir do endereço base 12, como em MIPS cada palavra corresponde a 4 bytes, podemos obter a posição $12/4 = 3$.

Logo, vamos representar nosso código em C, inicialmente com a operação acumuladora de soma, transformando diretamente, assumindo que as variáveis correspondem aos registradores iniciais, teremos:

```
f = f + g;
f = f + h;
f = f + i;
f = f + j;
```

Observe que, a seguinte operação também pode ser convertida em:

```
f = f + g + h + i + j;
```

Por fim, armazenamos em nosso Array A de posição 3, logo:

```
A[3] = f + g + h + i + j;
```

9

Considere o seguinte código em linguagem C:

```
f = f+A[2] + B[g];
```

Considere que as variáveis `f` e `g` estão memorizadas nos registradores `$s0` e `$s1`, respectivamente. Além disso, os endereços bases de `A` e `B` estão em `$s6` e `$s7`, respectivamente. Determine o código MIPS correspondente.

Resposta 9

Para montar nossas respostas, é necessário inicialmente acessar os valores que estão armazenados em Arrays, nesse caso, iremos armazenar temporariamente os Arrays `A[2]` e `B[g]`; logo:

```
lw $t0, 8($s6) # Carrega o valor de $s6 endereço 8 para $t0 (Carrega A[2] em t0)

add $t1, $s1, $s1 # $t1 = $s1 + $s1 (t1 = g + g = 2g)
add $t1, $t1, $t1 # $t1 = $t1 + $t1 (t1 = 2g + 2g = 4g)
add $t1, $s7, $t1 # $t1 = $t1 + $s7 (t1 = 4g + End. B)
lw $t1, 0($t1) # Carrega o valor de $t1 endereço 0 para $t1 (Carrega B[g] em t1)
```

Observe que, para obter o valor do Array `B` temos uma operação um pouco mais complexa, isso ocorre pelo fato de não conhecermos a posição `g` de imediato, logo é necessário preparar o endereço base, em MIPS cada palavra possui 4 bytes, logo precisamos multiplicar a posição por 4 bytes para obter o endereço da posição desejada. Por fim, é efetuada a operação de soma ao endereço base de `B` antes de armazenar a palavra (já que não sabemos o valor constante de `g`).

Por fim, desejamos realizar nossa operação de soma acumuladora, logo nosso código será:

```
# Carregamento dos Arrays
lw $t0, 8($s6) # Carrega o valor de $s6 endereço 8 para $t0 (Carrega A[2] em t0)
```

```

add $t1, $s1, $s1 # $t1 = $s1 + $s1 (t1 = g + g = 2g)
add $t1, $t1, $t1 # $t1 = $t1 + $t1 (t1 = 2g + 2g = 4g)
add $t1, $s7, $t1 # $t1 = $t1 + $s7 (t1 = 4g + End. B)
lw $t1, 0($t1) # Carrega o valor de $t1 endereço 0 para $t1 (Carrega B[g]
em t1)

# Operação de Soma
add $s0, $s0, $t0
add $s0, $s0, $t1

```

10

Considere o seguinte código em linguagem C:

```
f = f + A[B[g] + 2];
```

Considere que as variáveis f e g estão memorizadas nos registradores $\$s0$ e $\$s1$, respectivamente. Além disso, os endereços bases de A e B estão em $\$s6$ e $\$s7$, respectivamente. Determine o código MIPS correspondente.

Resposta 10

```

# Carrega B[g]
add $t1, $s1, $s1 # $t1 = $s1 + $s1 (t1 = g + g = 2g)
add $t1, $t1, $t1 # $t1 = $t1 + $t1 (t1 = 2g + 2g = 4g)
add $t1, $s7, $t1 # $t1 = $t1 + $s7 (t1 = 4g + End. B)
lw $t1, 0($t1) # Carrega o valor de $t1 endereço 0 para $t1 (Carrega B[g]
em t1)

# Carrega A[B[g] + 2]
addi $t1, $t1, 2
add $t1, $t1, $t1
add $t1, $t1, $t1
add $t1, $s6, $t1
lw $t1, 0($t1)

# Adiciona f = f + A[B[g] + 2]
add $s0, $s0, $t1

```

11

Considerando que o processador MIPS possui os seguintes valores em seus registradores e memória:

Registrador	Valor	Endereços de Memória	Valor
-------------	-------	----------------------	-------

Registrador	Valor	Endereços de Memória	Valor
\$s0	120	120	4
\$s1	28	128	25
\$s2	50	192	128
\$s3	192	208	32

Após a execução do seguinte código, qual o valor do registrador **\$s2** e do endereço de memória **120** ?

```
lw $t0, 8($s0)
sll $t1, $t0, 2
lw $t2, 16($s3)
slt $t7, $t1, $t2
beq $t7, $0, branch

add $s2, $s2, $s1
sw $s2, 8($s0)
j exit

branch:
    add $s2, $s2, $s1
    sw $s2, 0($s0)

exit:
```

Resposta 11

Vamos executar esse código por linha, logo:

```
lw $t0, 8($s0) # $t0 = 25 (Valor do endereço de memória $s0 (120) + 8 =
128)
sll $t1, $t0, 2 # $t1 = 25 * 2^2 = 100
lw $t2, 16($s3) # $t2 = 32 (Valor do endereço de memória $s3 (192) + 16
= 208)
slt $t7, $t1, $t2 # $t7 = 0 (pois $t1 não é menor que $t2)
beq $t7, $0, branch # $t7 é igual à 0, logo pula para o endereço da
label branch

add $s2, $s2, $s1 # Não executado
sw $s2, 8($s0) # Não executado
j exit # Não executado

branch:
    add $s2, $s2, $s1 # $s2 = 50 + 28 = 78
    sw $s2, 0($s0) # End. Memória 120 = 78

exit:
```

Logo o valor do Endereço de memória 120 e do registrador $\$s2$ serão ambos 78.