



Universidade Federal Rural do Semiárido
Departamento de Ciências Exatas e Naturais
Ciência da Computação
Prof. Sílvio Fernandes

1.3 – Lista de Exercícios de Arquitetura de Computadores – Assembly MIPS

1. Utilize o código a seguir para executar no MARS e responder as questões observando as informações desse simulador, executando passo-a-passo.

```
#####
#Funcao principal que chama a funcao folha
#####
.data
i: .word 1      # valor 5 armazenado em i
j: .word 1      # valor 2 armazenado em j
f: .word 0      # valor 1 armazenado em f
g: .word 4      # valor 3 armazenado em g
h: .word 4      # valor 0 armazenado em h
teste: .word 50
msg_result: .asciiz "O valor de F é: "
msg_teste: .asciiz "\nO valor da variavel teste é: "

.text
main:
    # carregando variaveis
    la $t0, teste      # endereco de "teste" em $t0
    lw $s0, 0($t0)     # carrega o valor de "teste" em $s0
    la $t0, g          # endereco de "g" em $t0
    lw $a0, 0($t0)     # carrega o valor de "g" em $a0
    la $t0, h          # endereco de "h" em $t0
    lw $a1, 0($t0)     # carrega o valor de "h" em $a1
    la $t0, i          # endereco de "i" em $t0
    lw $a2, 0($t0)     # carrega o valor de "i" em $a2
    la $t0, j          # endereco de "j" em $t0
    lw $a3, 0($t0)     # carrega o valor de "j" em $a3

    #chama a funcao
    jal exemplo_folha

    #copia resultado para $s1
    add $s1, $zero, $v0

    #imprime o resultado
    la $a0, msg_result # endereco de "msg_result" em $a0
    li $v0, 4          # especifica o serviço de impressao de string
    syscall            # faz a chamada de system para imprimir a
string

    add $a0, $s1, $zero # resultado final
    li $v0, 1          # especifica o servido de impressao de
inteiros
    syscall            # imprime o valor de resultado
```

```

#imprime variavel teste para certificar que o valor foi
restaurado
la    $a0, msg_teste # endereco de "msg_teste" em $a0
li    $v0, 4          # especifica o servico de impressao de string
syscall                # faz a chamada de system para imprimir a
string

add    $a0, $s0, $zero # valor do teste
li    $v0, 1          # especifica o servico de impressao de
inteiros
syscall                # imprime o valor de resultado

# Terminando o programa
li    $v0, 10         # system call for exit
syscall                # we are out of here.

#####
# Funcao folha # Calcula f = (g+h)-(i+j)
#####
exemplo_folha:
# Liberando registradores
addi $sp, $sp, -12    # ajusta a pilha criando espaço para três itens
sw $t1, 8($sp)        #salva registradores $t1, $t0 e $s0
sw $t0, 4($sp)        # para ser usado depois
sw $s0, 0($sp)
add $t0, $a0, $a1      # $t0 = g+h
add $t1, $a2, $a3      # $t1 = i + j
sub $s0, $t0, $t1      # $s0 = (g+h) - (i + j), ou seja, f

#Copiando o valor de f para ser retornado
add $v0, $s0, $zero

#Restaurando os três valores antigos dos registradores
lw $s0, 0($sp)
lw $t0, 4($sp)
lw $t1, 8($sp)
addi $sp, $sp, 12

# E finalmente, o procedimento termina
jr $ra                # Desvia de volta à rotina que chamou

```

- Após compilar, quais os valores armazenados em \$t0, \$t1, \$s0, \$sp, \$ra e \$pc
- Informe quais os valores armazenados em \$t0, \$t1, \$s0, \$sp, \$ra e \$pc, antes de executar `jal exemplo_folha`
- Informe os valores de \$ra e \$pc após a chamada da função
- Informe os endereços de memória e os respectivos valores armazenados neles durante o armazenamento dos registradores na pilha dentro da função. E qual relação com o \$sp

- e. Informe os valores de \$ra e \$pc após o retorno da função
2. Suponha que a primeira função listada na tabela se chame first. Você deverá traduzir essas rotinhas de código C para o assembly MIPS.

I	<pre> int fib(int n){ if(n==0) return 0; else if (n==1) return 1; else fib(n-1) + fib(n-2); } </pre>
II	<pre> int positive (int a, int b){ if (addit(a,b) > 0) return 1; else return 0; } int addit (int a, int b){ return a+b; } </pre>

- a) Implemente os dois códigos C da tabela em assembly do MIPS. Qual é o número total de instruções MIPS necessárias para executar a função?
- b) As funções normalmente podem ser implementadas pelos compiladores “em linha”. Um função em linha é quando o corpo da função é copiado para o espaço do programa, permitindo que o overhead da chamada de função seja eliminado. Implemente um versão “em linha” dos códigos C, Qual é a redução no número total de instruções assembly do MIPS necessárias para completar a função? Suponha que a variável C n seja inicializada como 5.
- c) Para cada chamada de função, mostre o conteúdo da pilha após a chamada de função ser feita. Suponha que o ponteiro de pilha esteja originalmente no endereço 9x7fffffc e siga as convenções de registrador especificadas.
3. Considere os 2 códigos a seguir

I	<pre> f: add \$v0, \$a1, \$a0 bnez \$a2, L sub \$v0, \$a0, \$a1 L: jr \$v0 </pre>
II	<pre> f: add \$a2, \$a3, \$a2 </pre>

	slt \$a2, \$a2, \$a0 move \$v0, \$a1 beqz \$a2, L jr \$ra L: move \$a0, \$a1 jal g
--	---

- Estes códigos contêm erro que viola a convenção de chamada do MIPS. Qual é o erro e como ele deve ser consertado?
 - Qual é o código equivalente C desses códigos assembly? Suponha que os argumentos da função se chamem a, b, c etc. na versão C da função.
 - No ponto em que a função é chamada, os registradores \$a0, \$a1, \$a2 e \$a3 têm os valores 1, 100, 1000 e 30, respectivamente. Qual é o valor retornando por essa função? Se outra função g for chamada a partir de f, suponha que o valor retornado de g sempre seja 500.
4. Para os problemas seguintes, você estará usando os dados binários dessa tabela:

I	0010 0000 0000 0001 0100 1001 0010 0100 ₂
II	0000 1111 1011 1110 0100 0000 0000 0000 ₂

- Escreva o código MIPS que cria as constantes de 32 bits listadas anteriormente e armazena esse valor no registrador \$t1
- Se o valor atual do PC for 0x0000 0000, você pode usar uma única instrução de salto para chegar ao endereço do PC, conforme na tabela anterior?