Exercício 08

Nome do aluno:

Fabio Volkmann Coelho

Objetivo

Consolidar o aprendizado da linguagem Assembly RISC-V e compreender como as instruções de suporte a procedimentos são executadas.

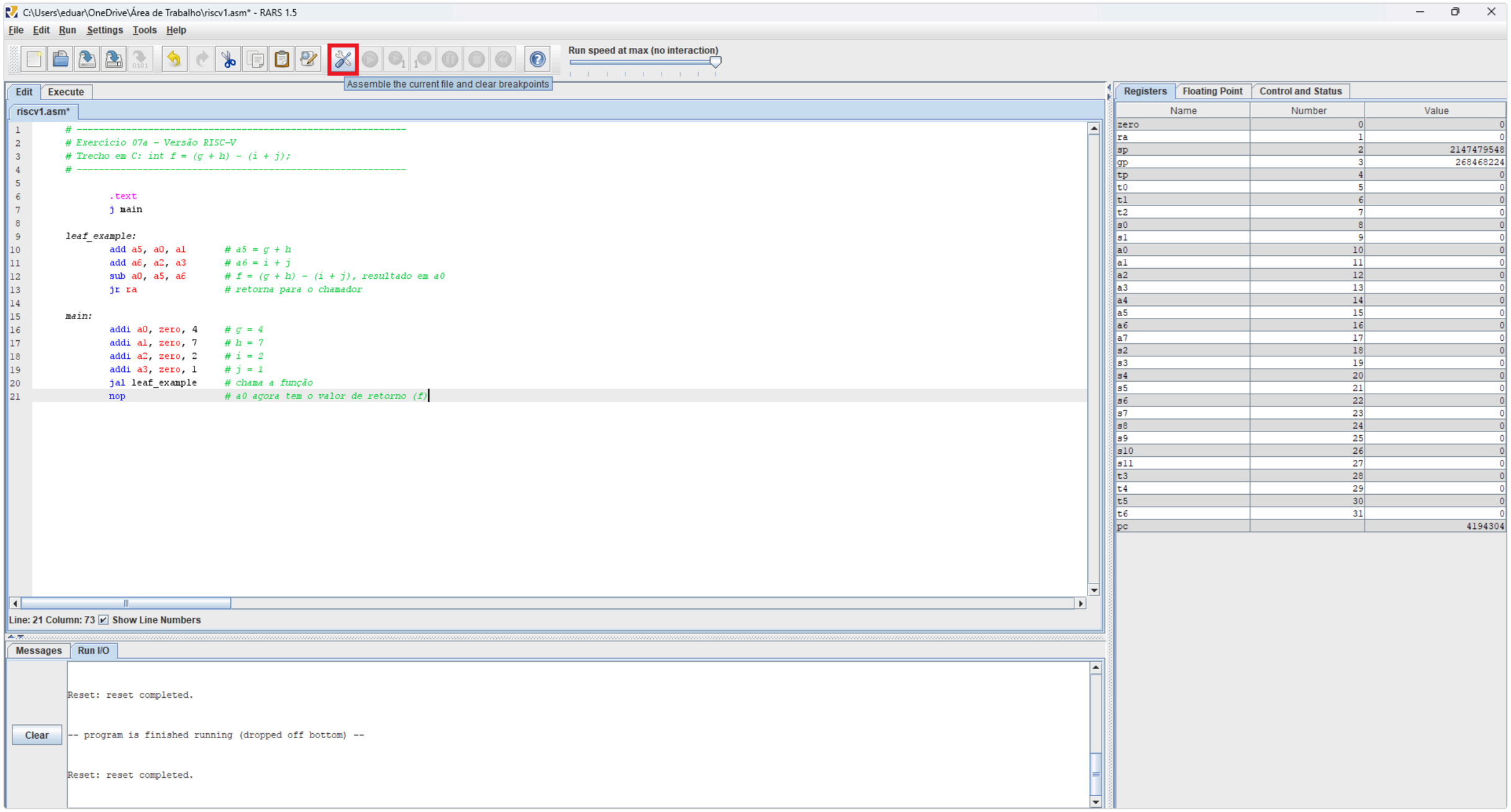
Instruções

1. Abra o simulador de linguagem RISC-V.
2. No editor de texto do simulador, transcreva o código abaixo:

```
# ------------------------------------------------------------
# Exercício 08 - Versão RISC-V
# Trecho em C: int f = (g + h) - (i + j);
# ------------------------------------------------------------

        .text
        j main

leaf_example:
        addi sp, sp, -12      # ajusta a pilha para 3 registradores
        sw   t1, 8(sp)        # salva t1
        sw   t0, 4(sp)        # salva t0
        sw   s0, 0(sp)        # salva s0

        add  t0, a1, a2       # t0 = g + h
        add  t1, a3, a4       # t1 = i + j
        sub  s0, t0, t1       # f = t0 - t1
        add  a0, s0, zero     # retorno em a0

        lw   s0, 0(sp)        # restaura s0
        lw   t0, 4(sp)        # restaura t0
        lw   t1, 8(sp)        # restaura t1
        addi sp, sp, 12       # libera espaço na pilha
        jr   ra               # retorna da função

main:
        addi t1, zero, 1
        addi t0, zero, 2
        addi s0, zero, 3

        addi a1, zero, 4      # g
        addi a2, zero, 7      # h
        addi a3, zero, 2      # i
        addi a4, zero, 1      # j

        jal leaf_example      # chama leaf_example
        nop                   # resultado em a0
```

Montagem e Execução

Clique no botão **Assemble** para montar o programa.



Faça a execução passo-a-passo do programa e, a cada instrução, preencha a tabela abaixo cada vez que o valor de um registrador ou posição da memória de dados for modificado.

Antes da execução da instrução		**Depois da execução da instrução**												
PC	**Instrução**	**f**	**g**	**h**	**i**	**j**						**Segmento de Dados (Pilha - sp)**		
		R10	**R11**	**R12**	**R13**	**R14**	**R1**	**R2**	**R5**	**R6**	**R8**			
		(a0)	**(a1)**	**(a2)**	**(a3)**	**(a4)**	**(ra)**	**(sp)**	**(t0)**	**(t1)**	**(s0)**	**0x7FFFEFF0**	**0x7FFFEFF4**	**0x7FFFEFF8**
		0x00000000	**0x00000000**	**0x00000000**	**0x00000000**	**0x00000000**	**0x00000000**	**0x7FFFEFFC**	**0x00000000**	**0x00000000**	**0x00000000**	**0x00000000**	**0x00000000**	**0x00000000**
0x00400000	j main													
0x00400038	addi t1, zero, 1									0x00000001				

Antes da execução da instrução		Depois da execução da instrução												
0x00400040	addi t0, zero, 2								0x00000002					
0x00400044	addi s0, zero, 3										0x00000003			
0x00400048	addi a1, zero, 4		0x00000004											
0x0040004C	addi a2, zero, 7			0x00000007										
0x00400050	addi a3, zero, 2				0x00000002									
0x00400054	addi a4, zero, 1					0x00000001								
0x00400058	jal leaf_example						0x00400058	0x7fffeffc						
0x00400004	addi sp, sp, -12							0x7fffeff0						
0x00400008	sw t1, 8(sp)													0x00000001
0x0040000C	sw t0, 4(sp)												0x00000002	
0x00400020	sw s0, 0(sp)											0x00000003		
0x00400024	add t0, a1, a2								0x0000000B					
0x00400028	add t1, a3, a4									0x00000003				
0x0040002C	sub s0, t0, t1										0x00000008			
0x00400030	add a0, s0, zer	0x00000008												
0x00400034	lw s0, 0(sp)										0x00000003			
0x00400038	lw t0, 4(sp)									0x00000001				
0x0040003C	lw t1, 8(sp)								0x00000002					
0x00400040	addi sp, sp, 12							0x7fffeffc						
0x00400044	jr ra													
0x00400060	nope	0x00000008	0x00000004	0x00000007	0x00000002	0x00000001	0x00400058	0x7fffeffc	0x00000002	0x00000001	0x00000003	0x00000003	0x00000002	0x00000001

OBS: Salve o PDF em formato A2 e Paisagem para garantir que todas as informações da página fiquem visíveis

Adicionar linha Salvar como PDF

Se desejar reiniciar o programa, clique no botão **Reset**.

C:\Users\eduar\OneDrive\Área de Trabalho\riscv1.asm - RARS 1.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute | Reset memory and registers

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x00400313	addi x6,x0,4	10: addi t1, zero, 4 # g = 4
	0x00400004	0x00300393	addi x7,x0,3	11: addi t2, zero, 3 # h = 3
	0x00400008	0x00200e13	addi x28,x0,2	12: addi t3, zero, 2 # i = 2
	0x0040000c	0x00100e93	addi x29,x0,1	13: addi t4, zero, 1 # j = 1
	0x00400010	0x00730f33	add x30,x6,x7	15: add t5, t1, t2 # t5 = g + h
	0x00400014	0x01de0fb3	add x31,x28,x29	16: add t6, t3, t4 # t6 = i + j
	0x00400018	0x41ff02b3	sub x5,x30,x31	17: sub t0, t5, t6 # f = t5 - t6 (resultado em t0)

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

Messages Run I/O

```
Reset: reset completed.
```

Clear

Registers Floating Point Control and Status

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7fffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000004
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400004

← Voltar ao tutorial