

## Exercício 01

### Objetivo:

Aprender a utilizar a interface do RARS executando o primeiro exemplo de programação na linguagem de montagem do RISC-V no livro texto.

### Instruções:

1. Inicie o RARS.
2. No editor de textos do RARS, transcreva o código abaixo e salve o arquivo com o nome **exercicio\_01**.

```
#####
# Exercício 01 - Patterson pags. 54/55/56
# Mostra a compilação de um comando de atribuição em C
#####
# Trecho em C:
#
# f = (g + h) - (i + j)

        .text    # segmento de código (programa)

main:
    addi s1, zero, 4
    addi s2, zero, 3
    addi s3, zero, 2
    addi s4, zero, 1

    add t0, s1, s2    # t0 = g + h
    add t1, s3, s4    # t1 = i + j
    sub s0, t0, t1    # f = t0 - t1
```

3. Para iniciar a montagem do código vá ao menu **Run** e selecione a opção **Assemble** ou pressione **F3**.
4. Observe que é destacada a primeira instrução do segmento de texto (código): `addi s1, zero, 4`. O endereço dessa instrução é o `0x00400000` (em hexadecimal). Ao lado do endereço são mostradas outras informações a serem explicadas logo a seguir.

Text Segment				
Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x00400493	<code>addi x9,x0,4</code>	12: <code>addi s1, zero, 4</code>
<input type="checkbox"/>	0x00400004	0x00300913	<code>addi x18,x0,3</code>	13: <code>addi s2, zero, 3</code>
<input type="checkbox"/>	0x00400008	0x00200993	<code>addi x19,x0,2</code>	14: <code>addi s3, zero, 2</code>
<input type="checkbox"/>	0x0040000c	0x00100a13	<code>addi x20,x0,1</code>	15: <code>addi s4, zero, 1</code>
<input type="checkbox"/>	0x00400010	0x012482b3	<code>add x5,x9,x18</code>	17: <code>add t0, s1, s2</code> # t0 = g + h
<input type="checkbox"/>	0x00400014	0x01498333	<code>add x6,x19,x20</code>	18: <code>add t1, s3, s4</code> # t1 = i + j
<input type="checkbox"/>	0x00400018	0x40628433	<code>sub x8,x5,x6</code>	19: <code>sub s0, t0, t1</code> # f = t0 - t1

Observe que, da esquerda para direita, são mostrados:

- opção para inserção de *breakpoints*;
- endereço da instrução (em hexadecimal);
- código de máquina da instrução (em hexadecimal);
- código assembly da instrução com registradores identificados pelos seus endereços;
- código assembly da instrução com registradores identificados pelos seus nomes (código originalmente escrito).

5. Observe na janela de registradores o conteúdo do registrador PC. Note que o seu valor é igual ao do endereço da instrução que será executada, mostrada anteriormente.

s10	26	0x0000000000000000
s11	27	0x0000000000000000
t3	28	0x0000000000000000
t4	29	0x0000000000000000
t5	30	0x0000000000000000
t6	31	0x0000000000000000
pc		0x0000000000004000

6. Observe o conteúdo do registrador **9 (s1)** na janela de registradores. Pressione a tecla **F7** e note que o seu conteúdo muda de **00000000** para **00000004** devido à execução da instrução, conforme destacado abaixo:

#### Antes

t0	5	0x0000000000000000
t1	6	0x0000000000000000
t2	7	0x0000000000000000
s0	8	0x0000000000000000
s1	9	0x0000000000000000
a0	10	0x0000000000000000
a1	11	0x0000000000000000

#### Depois

t0	5	0x0000000000000000
t1	6	0x0000000000000000
t2	7	0x0000000000000000
s0	8	0x0000000000000000
s1	9	0x0000000000000004
a0	10	0x0000000000000000
a1	11	0x0000000000000000

7. A execução dessa primeira instrução modificou apenas o conteúdo do registrador s1. As demais instruções irão modificar outros registradores. Execute o programa passo-a-passo e preencha a tabela a seguir, marcando o conteúdo do registrador PC, a instrução a ser executada e o valor dos registradores após a execução da instrução. Anote o valor de um registrador apenas quando ele é modificado.

Antes		Depois da execução da instrução						
PC	Instrução	R8	R9	R16	R17	R18	R19	R20
		(\$t0)	(\$t1)	(\$s0)	(\$s1)	(\$s2)	(\$s3)	(\$s4)
		00000000	00000000	00000000	00000000	00000000	00000000	00000000
00400000	addi s1,zero,4				00000004			

NOTA: Se for necessário reiniciar o programa, faça: **Run > Reset**.