



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**THAYANE STHEFFANY SILVA BARROS**

**ROTEIRO 7**

# Problema 1 - Código em Assembly (Fatorial de N)

Implementar em Assembly - RISC-V, um programa que calcula o fatorial de um número N. O valor de N deverá ser "carregado" em algum registrador de uso genérico no início da execução do programa.

---

Editor

Simulator

```
1 .data
2     N: .word 3
3
4 .text
5     .globl main
6
7 main:
8     lw a0, N
9     jal ra, factorial
10
11 factorial:
12     addi sp, sp, -4
13     sw ra, 0(sp)
14
15     li t0, 1
16
17 loop:
18     beqz a0, fim
19     mul t0, t0, a0
20     addi a0, a0, -1
21     j loop
22
23 fim:
24     mv a0, t0
25
26     lw ra, 0(sp)
27     addi sp, sp, 4
28
29     nop
```

- Carrega o valor da variável N no registrador a0. Isso é feito usando a instrução lw (load word).

Valores em hexadecimal      Valores em decimal

Registers	Memory
zero	0x00000000
ra (x1)	0x00000000
sp (x2)	0xfffffffff0
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000000
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000003
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Registers	Memory
zero	0
ra (x1)	0
sp (x2)	2147483632
gp (x3)	268435456
tp (x4)	0
t0 (x5)	0
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	0
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

- Registrador a0 é carregado com o valor N, que nesse exemplo é 3 (lw a0, N ).

Registers	Memory
zero	0x00000000
ra (x1)	0x00000000
sp (x2)	0xfffffffff0
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000000
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x10000000
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Registers	Memory
zero	0
ra (x1)	0
sp (x2)	2147483632
gp (x3)	268435456
tp (x4)	0
t0 (x5)	0
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	268435456
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

- Faz uma chamada para a função factorial, armazenando o endereço de retorno no registrador ra (return address register), que receberá o valor 12.

Registers	Memory
zero	0x00000000
ra (x1)	0x00000000
sp (x2)	0xfffffffff0
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000000
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000003
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Registers	Memory
zero	0
ra (x1)	0
sp (x2)	2147483632
gp (x3)	268435456
tp (x4)	0
t0 (x5)	0
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	3
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

- Chama a função factorial. Nessa parte do código abaixo é reservado um espaço na pilha para armazenar o registrador de retorno.

The screenshot shows the RISC-V Simulator interface with the following sections:

- Machine Code:** A table of assembly instructions and their binary representations.
- Basic Code:** The assembly code corresponding to the machine code.
- Original Code:** The assembly code as it was initially input.
- Registers:** A table showing the current values of all general-purpose registers (zero, ra, sp, gp, tp, t0, t1, t2, s0, s1, a0, a1, a2, a3).
- Memory:** A table showing the current values at memory addresses 0x00000000 through 0x0000000f.
- console output:** A text area showing the command-line interface history.

- Salva o valor do registrador `ra` na pilha. Isso é necessário para preservar o valor durante a execução da função.

The screenshot shows the RISC-V Simulator interface with the following sections:

- Editor**: Contains buttons for Run, Step, Prev, Reset, and Dump.
- Machine Code**: A table showing memory addresses (0x10000517, 0x00052503, etc.) and their corresponding assembly instructions (auipc, lw, jal, addi).
- Basic Code**: A table showing memory addresses (0x00112923, 0x00100293, etc.) and their corresponding assembly instructions (sw, addi, beq, mul, etc.). The row for address 0x00112923 is highlighted in green.
- Original Code**: A table showing memory addresses (0xffff50513, 0xff5ff06f, etc.) and their corresponding assembly instructions (addi, jal, mv).
- Registers**: A table showing register values for zero, ra (x1), sp (x2), gp (x3), tp (x4), t0 (x5), t1 (x6), t2 (x7), s0 (x8), s1 (x9), a0 (x10), a1 (x11), a2 (x12), and a3 (x13). The row for sp (x2) is highlighted in blue.
- Memory**: A table showing memory contents at various addresses.
- console output**: A text area containing the instruction being executed: "sw x1 0(x2)".
- Display Format**: Buttons for Hex, Decimal, and Octal.

- Inicializa a variável t0 (temporária) com o valor 1. Essa variável será usada para armazenar o resultado do fatorial. (li t0, 1)

The screenshot shows a debugger interface with several panes:

- Editor** tab: Contains the assembly code listing.
- Simulator** tab: Active, showing the state of registers and memory.
- Run**, **Step**, **Prev**, **Reset**, **Dump** buttons.
- Machine Code**, **Basic Code**, **Original Code** columns.
- Registers** pane: Shows register values for x0 to x13. The value for **sp (x2)** is highlighted in blue.
- Memory** pane: Shows memory starting at address 0x00000000.
- Console output**: Displays the command "j loop".

- Verifica se o valor de N (armazenado em a0) é igual a zero. Como não são ( $a0 = 3$ ) o programa continua a execução.

Registers Memory

zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0x7fffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000001
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000003
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Display Settings Hex

Registers Memory

zero	0
ra (x1)	12
sp (x2)	2147483628
gp (x3)	268435456
tp (x4)	0
t0 (x5)	1
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	3
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

Display Settings Decimal

- Multiplica o valor atual do factorial (t0) pelo valor de N (a0). O resultado é armazenado novamente em t0. nesse exemplo vai receber o valor 3

Registers Memory

zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0x7fffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000001
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000003
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Display Settings Hex

Registers Memory

zero	0
ra (x1)	12
sp (x2)	2147483628
gp (x3)	268435456
tp (x4)	0
t0 (x5)	1
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	3
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

Display Settings Decimal

- Decrementa o valor de N em 1 (addi a0, a0, -1). a0 agora é carregado com valor 2.

Registers Memory

zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0x7fffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000003
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000003
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Display Settings Hex

Registers Memory

zero	0
ra (x1)	12
sp (x2)	2147483628
gp (x3)	268435456
tp (x4)	0
t0 (x5)	3
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	3
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

Display Settings Decimal

- Salta para a marcação loop, reiniciando o loop.

The screenshot shows the CPU simulator interface. The left pane displays the assembly code:

```

Machine Code          Basic Code           Original Code
0x10000517          auipc x10 65536    lw a0, N
0x00052503          lw x10 0(x10)      lw a0, N
0x004000ef          jal x1 4          jal ra, factorial
0xffffc10113         addi x2 x2 -4     addi sp, sp, -4
0x00112023         sw x1 0(x2)       sw ra, 0(sp)
0x00100293         addi x5 x0 1       li t0, 1
0x00050863         beq x10 x0 16     beqz a0, fim
0x02a282b3          mul x5 x5 x10      mul t0, t0, a0
0xffff50513         addi x10 x10 -1    addi a0, a0, -1
0xffff5ff06f         jal x0 -12        j loop
0x00028513          addi x10 x5 0       mv a0, t0

```

The right pane shows the Registers and Memory sections. The Registers section lists the values of various processor registers. The Memory section shows the contents of memory locations, including the value of `a0` which is currently 2.

- Verifica se o valor de N (armazenado em a0) é igual a zero. Como não são ( $a0 = 2$ ) o programa continua a execução.

The screenshot shows the CPU simulator interface. The left pane displays the assembly code:

```

Machine Code          Basic Code           Original Code
0x10000517          auipc x10 65536    lw a0, N
0x00052503          lw x10 0(x10)      lw a0, N
0x004000ef          jal x1 4          jal ra, factorial
0xffffc10113         addi x2 x2 -4     addi sp, sp, -4
0x00112023         sw x1 0(x2)       sw ra, 0(sp)
0x00100293         addi x5 x0 1       li t0, 1
0x00050863         beq x10 x0 16     beqz a0, fim
0x02a282b3          mul x5 x5 x10      mul t0, t0, a0
0xffff50513         addi x10 x10 -1    addi a0, a0, -1
0xffff5ff06f         jal x0 -12        j loop
0x00028513          addi x10 x5 0       mv a0, t0

```

The right pane shows the Registers and Memory sections. The Registers section lists the values of various processor registers. The Memory section shows the contents of memory locations, including the value of `a0` which is currently 2.

- Multiplica o valor atual do factorial (t0) pelo valor de N (a0). O resultado é armazenado novamente em t0. nesse exemplo vai receber o valor 6 ( $3 * 2$ ).

The screenshot shows the CPU simulator interface. The left pane displays the assembly code:

```

Machine Code          Basic Code           Original Code
0x10000517          auipc x10 65536    lw a0, N
0x00052503          lw x10 0(x10)      lw a0, N
0x004000ef          jal x1 4          jal ra, factorial
0xffffc10113         addi x2 x2 -4     addi sp, sp, -4
0x00112023         sw x1 0(x2)       sw ra, 0(sp)
0x00100293         addi x5 x0 1       li t0, 1
0x00050863         beq x10 x0 16     beqz a0, fim
0x02a282b3          mul x5 x5 x10      mul t0, t0, a0
0xffff50513         addi x10 x10 -1    addi a0, a0, -1
0xffff5ff06f         jal x0 -12        j loop
0x00028513          addi x10 x5 0       mv a0, t0

```

The right pane shows the Registers and Memory sections. The Registers section lists the values of various processor registers. The Memory section shows the contents of memory locations, including the value of `a0` which is currently 2.

- Decrements the value of N by 1 (addi a0, a0, -1). Now it is loaded with the value 1.

Registers

zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0xffffffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000000
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000002
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Memory

zero	0
ra (x1)	12
sp (x2)	2147483628
gp (x3)	268435456
tp (x4)	0
t0 (x5)	6
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	2
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

- Jumps to the loop marking, restarting the loop.

Registers

zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0xffffffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000006
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000001
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Memory

zero	0
ra (x1)	12
sp (x2)	2147483628
gp (x3)	268435456
tp (x4)	0
t0 (x5)	6
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	1
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

- Checks if the value of N (stored in a0) is equal to zero. Since it's not (a0 = 1), the program continues execution.

Registers

zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0xffffffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000006
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000001
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Memory

zero	0
ra (x1)	12
sp (x2)	2147483628
gp (x3)	268435456
tp (x4)	0
t0 (x5)	6
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	1
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

- Multiplica o valor atual do fatorial (t0) pelo valor de N (a0). O resultado é armazenado novamente em t0. nesse exemplo vai receber o valor 6 ( $3 \times 2 \times 1$ ).

The screenshot shows the assembly code for calculating factorial. The highlighted instruction at address 0x02a282b3 is `mul t0, t0, a0`. The registers and memory pane show the state of the processor after this step.

Registers	Memory
zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0x7fffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000006
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000001
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

- Decrementa o valor de N em 1 (addi a0, a0, -1). a0 agora é carregado com valor 0.

The screenshot shows the assembly code for calculating factorial. The highlighted instruction at address 0xffff50513 is `addi a0, a0, -1`. The registers and memory pane show the state of the processor after this step.

Registers	Memory
zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0x7fffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000006
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000001
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

- Salta para a marcação loop, reiniciando o loop.

The screenshot shows the assembly code for calculating factorial. The highlighted instruction at address 0xffff5ff06f is `jal x0 -12`, which is the jump to the loop label. The registers and memory pane show the state of the processor after this step.

Registers	Memory
zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0x7fffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000006
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000000
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

- Verifica se o valor de N (armazenado em a0) é igual a zero. Como o valor é igual, desvia para a marcação fim e encerra a função.

Machine Code Basic Code Original Code

0x10000517	auipc x10 65536	lw a0, N
0x00052503	lw x10 0(x10)	lw a0, N
0x004000ef	jal x1 4	jal ra, factorial
0xffffc10113	addi x2 x2 -4	addi sp, sp, -4
0x00112023	sw x1 0(x2)	sw ra, 0(sp)
0x00100293	addi x5 x0 1	li t0, 1
0x00050863	beq x10 x0 16	beqz a0, fim
0x02a282b3	mul x5 x5 x10	mul t0, t0, a0
0xffff50513	addi x10 x10 -1	addi a0, a0, -1
0xffff5ff06f	jal x0 -12	j loop
0x00028513	addi x10 x5 0	mv a0, t0

Registers Memory

zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0x7fffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000006
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000000
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Registers Memory

zero	0
ra (x1)	12
sp (x2)	2147483628
gp (x3)	268435456
tp (x4)	0
t0 (x5)	0
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	0
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

- Move o valor do factorial (t0) para o registrador a0, que será o valor retornado pela função.

Machine Code Basic Code Original Code

0x10000517	auipc x10 65536	lw a0, N
0x00052503	lw x10 0(x10)	lw a0, N
0x004000ef	jal x1 4	jal ra, factorial
0xffffc10113	addi x2 x2 -4	addi sp, sp, -4
0x00112023	sw x1 0(x2)	sw ra, 0(sp)
0x00100293	addi x5 x0 1	li t0, 1
0x00050863	beq x10 x0 16	beqz a0, fim
0x02a282b3	mul x5 x5 x10	mul t0, t0, a0
0xffff50513	addi x10 x10 -1	addi a0, a0, -1
0xffff5ff06f	jal x0 -12	j loop
0x00028513	addi x10 x5 0	mv a0, t0

Registers Memory

zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0x7fffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000006
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000000
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Registers Memory

zero	0
ra (x1)	12
sp (x2)	2147483628
gp (x3)	268435456
tp (x4)	0
t0 (x5)	6
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	0
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

- Recupera o valor do registrador de retorno (ra) da pilha.

Machine Code Basic Code Original Code

0x10000517	auipc x10 65536	lw a0, N
0x00052503	lw x10 0(x10)	lw a0, N
0x004000ef	jal x1 4	jal ra, factorial
0xffffc10113	addi x2 x2 -4	addi sp, sp, -4
0x00112023	sw x1 0(x2)	sw ra, 0(sp)
0x00100293	addi x5 x0 1	li t0, 1
0x00050863	beq x10 x0 16	beqz a0, fim
0x02a282b3	mul x5 x5 x10	mul t0, t0, a0
0xffff50513	addi x10 x10 -1	addi a0, a0, -1
0xffff5ff06f	jal x0 -12	j loop
0x00028513	addi x10 x5 0	mv a0, t0
0x00012083	lw x1 0(x2)	lw ra, 0(sp)

Registers Memory

zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0x7fffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000006
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000006
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Registers Memory

zero	0
ra (x1)	12
sp (x2)	2147483628
gp (x3)	268435456
tp (x4)	0
t0 (x5)	6
t1 (x6)	0
t2 (x7)	0
s0 (x8)	0
s1 (x9)	0
a0 (x10)	6
a1 (x11)	0
a2 (x12)	0
a3 (x13)	0

- Libera o espaço reservado na pilha.

Registers Memory

zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0x7fffffec
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000006
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000006
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Memory

Address	+0	+1	+2	+3
0x7fffffd4	00	00	00	00
0x7fffffd8	00	00	00	00
0x7fffffdc	00	00	00	00
0x7fffffe0	00	00	00	00
0x7fffffe4	00	00	00	00
0x7fffffe8	00	00	00	00
0x7fffffec	0c	00	00	00
0x7fffffd0	00	00	00	00
0x7fffffd4	00	00	00	00

- A instrução nop não faz nada. Ela é usada como uma instrução vazia.

Registers Memory

zero	0x00000000
ra (x1)	0x0000000c
sp (x2)	0x7fffffd0
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000006
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000006
a1 (x11)	0x00000000
a2 (x12)	0x00000000
a3 (x13)	0x00000000

Memory

Address	+0	+1	+2	+3
0x7fffffd4	00	00	00	00
0x7fffffd8	00	00	00	00
0x7fffffdc	00	00	00	00
0x7fffffe0	00	00	00	00
0x7fffffe4	00	00	00	00
0x7fffffe8	00	00	00	00
0x7fffffec	0c	00	00	00
0x7fffffd0	00	00	00	00
0x7fffffd4	00	00	00	00

- Print da memória

Registers Memory

Address	+0	+1	+2	+3
0x7fffffd4	00	00	00	00
0x7fffffd8	00	00	00	00

Memory

Address	+0	+1	+2	+3
0x7fffffd4	00	00	00	00
0x7fffffd8	00	00	00	00

## Problema 2 - Leitura e Escrita na Memória de Dados

Implementar em Assembly - RISC-V, um programa que realiza as operações a seguir (IMPORTANTE: observar o uso das instruções "li" e "lui").

- Escrever o valor 0xFE no endereço da memória de dados 0x1000000C.
- Ler o valor 0xFE do endereço da memória de dados 0x1000000C e escrever no registrador x10.

Editor

Simulator

```
1 .data
2     valor: .word 0xFE
3
4 .text
5     .globl main
6
7 main:
8     lui t0, 0x10000
9     addi t0, t0, 0x000C
10    li t1, 0xFE
11    sw t1, 0(t0)
12
13    lui t0, 0x10000
14    addi t0, t0, 0x000C
15    lw x10, 0(t0)
16
17    li a7, 10
18    nop
19
```

- A instrução lui carrega os 20 bits mais significativos do endereço da memória de dados 0x1000000C no registrador t0.

The screenshot shows a CPU simulation interface. The 'Registers' window on the right displays the following register values:

Register	Value
zero	0x00000000
ra	0x00000000
(x1)	
sp	0x7fffffff0
(x2)	
gp	0x10000000
(x3)	
tp	0x00000000
(x4)	
t0	0x00000000
(x5)	
t1	0x00000000
(x6)	
t2	0x00000000
(x7)	

The 'Display' dropdown is set to 'Hex'.

- A instrução addi adiciona os 12 bits menos significativos do endereço da memória de dados 0x1000000C ao registrador t0.

The screenshot shows a CPU simulation interface. The 'Registers' window on the right displays the following register values:

Register	Value
zero	0x00000000
ra	0x00000000
(x1)	
sp	0x7fffffff0
(x2)	
gp	0x10000000
(x3)	
tp	0x00000000
(x4)	
t0	0x10000000
(x5)	
t1	0x00000000
(x6)	
t2	0x00000000
(x7)	

The 'Display' dropdown is set to 'Hex'.

- A instrução li carrega o valor 0xFE no registrador t1.

The screenshot shows a CPU simulation interface. The 'Registers' window on the right displays the following register values:

Register	Value
zero	0x00000000
ra	0x00000000
(x1)	
sp	0x7fffffff0
(x2)	
gp	0x10000000
(x3)	
tp	0x00000000
(x4)	
t0	0x1000000c
(x5)	
t1	0x00000000
(x6)	
t2	0x00000000
(x7)	

The 'Display' dropdown is set to 'Hex'.

- A instrução sw armazena o valor de t1 no endereço especificado por t0.

Registers

zero	0x00000000
ra	0x00000000
(x1)	
sp	0x7fffffff0
(x2)	
gp	0x10000000
(x3)	
tp	0x00000000
(x4)	
t0	0x1000000c
(x5)	
t1	0x000000fe
(x6)	
t2	0x00000000
(x7)	

Memory

0x1000000c	0x000000fe
------------	------------

- A instrução lui carrega novamente os 20 bits mais significativos do endereço da memória de dados 0x1000000C no registrador t0

Registers

zero	0x00000000
ra	0x00000000
(x1)	
sp	0x7fffffff0
(x2)	
gp	0x10000000
(x3)	
tp	0x00000000
(x4)	
t0	0x1000000c
(x5)	
t1	0x000000fe
(x6)	
t2	0x00000000
(x7)	

Memory

0x1000000c	0x0000000c
------------	------------

- A instrução addi adiciona novamente os 12 bits menos significativos do endereço da memória de dados 0x1000000C ao registrador t0.

Registers

zero	0x00000000
ra	0x00000000
(x1)	
sp	0x7fffffff0
(x2)	
gp	0x10000000
(x3)	
tp	0x00000000
(x4)	
t0	0x10000000
(x5)	
t1	0x000000fe
(x6)	
t2	0x00000000
(x7)	

Memory

0x1000000c	0x00000000
------------	------------

- A instrução lw carrega o valor do endereço especificado por t0 no registrador x10.

Machine Code    Basic Code    Original Code

0x100002b7	lui x5 65536	lui t0, 0x10000
0x00c28293	addi x5 x5 12	addi t0, t0, 0x000C
0x0fe00313	addi x6 x0 254	li t1, 0xFE
0x0062a023	sw x6 0(x5)	sw t1, 0(t0)
0x100002b7	lui x5 65536	lui t0, 0x10000
0x00c28293	addi x5 x5 12	addi t0, t0, 0x000C
<b>0x0002a503</b>	<b>lw x10 0(x5)</b>	<b>lw x10, 0(t0)</b>
0x00a00893	addi x17 x0 10	li a7, 10
0x000000013	addi x0 x0 0	nop

Registers

gp	0x10000000
(x3)	
tp	0x00000000
(x4)	
t0	0x1000000c
(x5)	
t1	0x000000fe
(x6)	
t2	0x00000000
(x7)	
s0	0x00000000
(x8)	
s1	0x00000000
(x9)	
a0	0x00000000
(x10)	
a1	0x00000000
(x11)	
a2	0x00000000
(x12)	
a3	0x00000000
(x13)	
a4	0x00000000
(x14)	
a5	0x00000000
(x15)	
a6	0x00000000
(x16)	
a7	0x00000000
(x17)	

Display Settings: Hex

- A instrução li carrega o código da chamada do sistema para encerrar o programa (10) no registrador a7

Machine Code    Basic Code    Original Code

0x100002b7	lui x5 65536	lui t0, 0x10000
0x00c28293	addi x5 x5 12	addi t0, t0, 0x000C
0x0fe00313	addi x6 x0 254	li t1, 0xFE
0x0062a023	sw x6 0(x5)	sw t1, 0(t0)
0x100002b7	lui x5 65536	lui t0, 0x10000
0x00c28293	addi x5 x5 12	addi t0, t0, 0x000C
<b>0x0002a503</b>	<b>lw x10 0(x5)</b>	<b>lw x10, 0(t0)</b>
<b>0x00a00893</b>	<b>addi x17 x0 10</b>	<b>li a7, 10</b>
0x000000013	addi x0 x0 0	nop

Registers

a0	0x000000fe
(x10)	
a1	0x00000000
(x11)	
a2	0x00000000
(x12)	
a3	0x00000000
(x13)	
a4	0x00000000
(x14)	
a5	0x00000000
(x15)	
a6	0x00000000
(x16)	
a7	0x00000000
(x17)	

Display Settings: Decimal

- A instrução nop não faz nada. Ela é usada como uma instrução vazia.

Machine Code    Basic Code    Original Code

0x100002b7	lui x5 65536	lui t0, 0x10000
0x00c28293	addi x5 x5 12	addi t0, t0, 0x000C
0x0fe00313	addi x6 x0 254	li t1, 0xFE
0x0062a023	sw x6 0(x5)	sw t1, 0(t0)
0x100002b7	lui x5 65536	lui t0, 0x10000
0x00c28293	addi x5 x5 12	addi t0, t0, 0x000C
<b>0x0002a503</b>	<b>lw x10 0(x5)</b>	<b>lw x10, 0(t0)</b>
<b>0x00a00893</b>	<b>addi x17 x0 10</b>	<b>li a7, 10</b>
<b>0x000000013</b>	<b>addi x0 x0 0</b>	<b>nop</b>

Registers

a0	0x000000fe
(x10)	
a1	0x00000000
(x11)	
a2	0x00000000
(x12)	
a3	0x00000000
(x13)	
a4	0x00000000
(x14)	
a5	0x00000000
(x15)	
a6	0x00000000
(x16)	
a7	0x0000000a
(x17)	

Display Settings: Hex

- Print da memoria

The screenshot shows a debugger interface with the following sections:

- Machine Code**: A table showing assembly instructions in machine code format.
- Basic Code**: A table showing assembly instructions in basic code format.
- Original Code**: A table showing assembly instructions in original code format.
- Registers**: A table showing the state of CPU registers.
- Memory**: A table showing the state of memory at various addresses.
- Console Output**: A text area containing the message "console output".
- Display Settings**: A dropdown menu set to "Decimal".

Machine Code	Basic Code	Original Code
0x100002b7	lui x5 65536	lui t0, 0x10000
0x00c28293	addi x5 x5 12	addi t0, t0, 0x00C
0x0fe00313	addi x6 x0 254	li t1, 0xFE
0x0002a023	sw x6 0(x5)	sw t1, 0(t0)
0x100002b7	lui x5 65536	lui t0, 0x10000
0x00c28293	addi x5 x5 12	addi t0, t0, 0x00C
0x0002a503	lw x10 0(x5)	lw x10, 0(t0)
0x00a00893	addi x17 x0 10	li a7, 10
0x00000013	addi x0 x0 0	nop

Address	+0	+1	+2	+3
0x10000024	0	0	0	0
0x10000020	0	0	0	0
0x1000001C	0	0	0	0
0x10000018	0	0	0	0
0x10000014	0	0	0	0
0x10000010	0	0	0	0
0x1000000C	-2	0	0	0
0x10000008	0	0	0	0
0x10000004	0	0	0	0
0x10000000	-2	0	0	0
0x0fffffc	0	0	0	0
0x0fffffb8	0	0	0	0
0x0fffff4	0	0	0	0

Registers:

Register	Value
a7	10
t0	0
t1	FE
x0	0
x10	0
x17	10
x5	0
x6	0

Memory Dump:

Address	Value
0x10000024	0
0x10000020	0
0x1000001C	0
0x10000018	0
0x10000014	0
0x10000010	0
0x1000000C	-2
0x10000008	0
0x10000004	0
0x10000000	-2
0x0fffffc	0
0x0fffffb8	0
0x0fffff4	0

Console Output:

```
console output
```

Display Settings: Decimal