**Lab 1**

15/07/2024

1. **Identificação dos participantes**

|  |  |  |
| --- | --- | --- |
| Nome | RGA | Turma |
| **Anthony Muniz Prado de Oliveira** | **202011722003** | **CC** |

1. **Introdução.**

Neste trabalho vocês exercitarão seus conhecimentos sobre a ISA (*Instruction Set Architecture*) do processador MIPS de 32 bits. Para isso, vocês construirão programas em Assembly que explorarão movimentação de dados entre o processador e a memória, bem como a construção de estruturas de controle de fluxo, tais como if-else, for e while. As implementações devem ser executadas no simulador MARS, a fim de verificar a corretude da implementação.

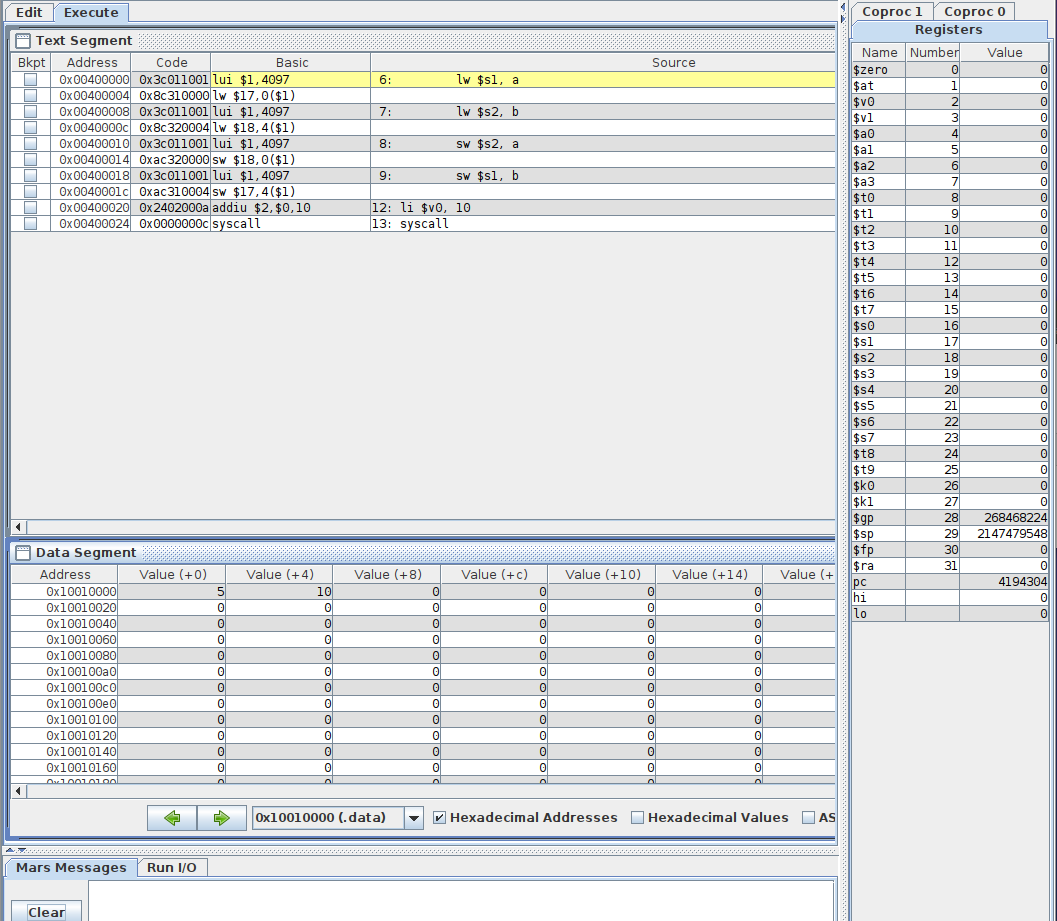
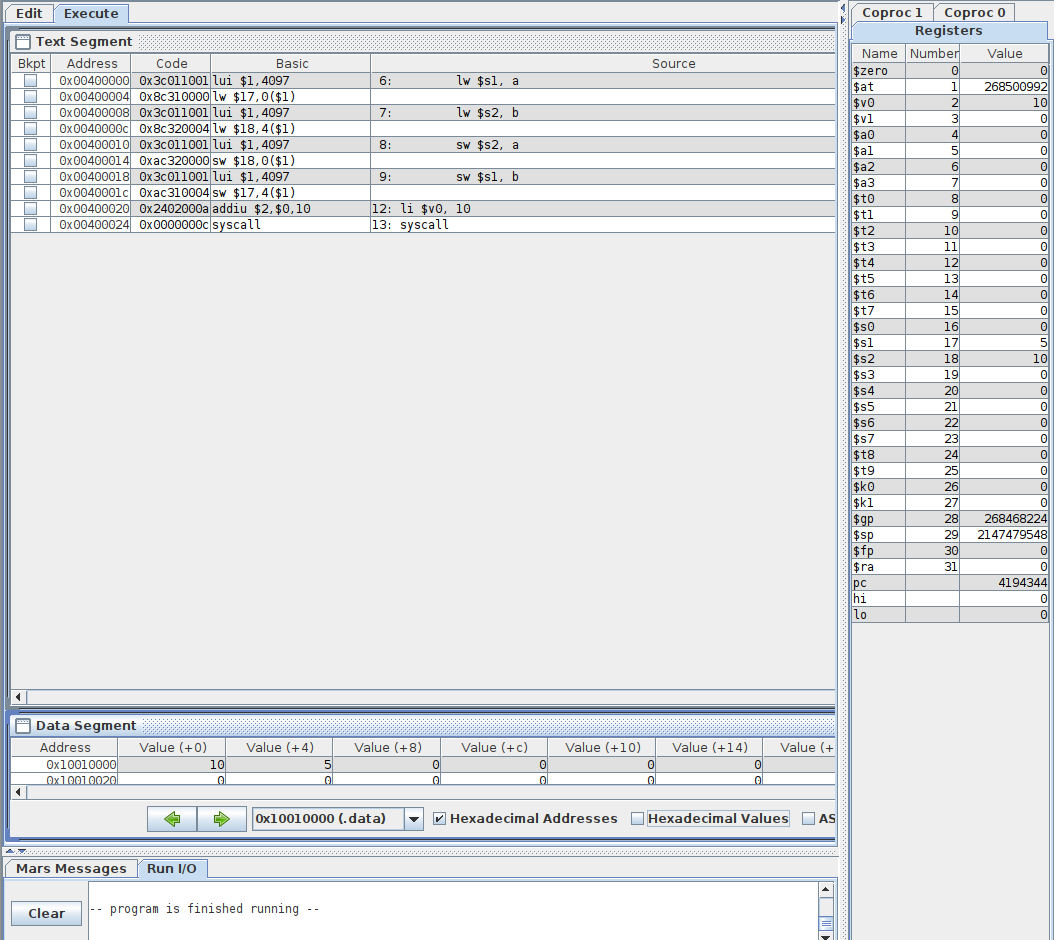
Para cada exercício, anexe o código fonte em um arquivo com o nome indicado, e cole um *print* da tela com o resultado final dado pelo simulador. O trabalho pode ser feito individualmente ou em dupla. Para o caso de duplas, apenas um dos alunos deve submeter o relatório no site da disciplina.

1. **Exercícios.**

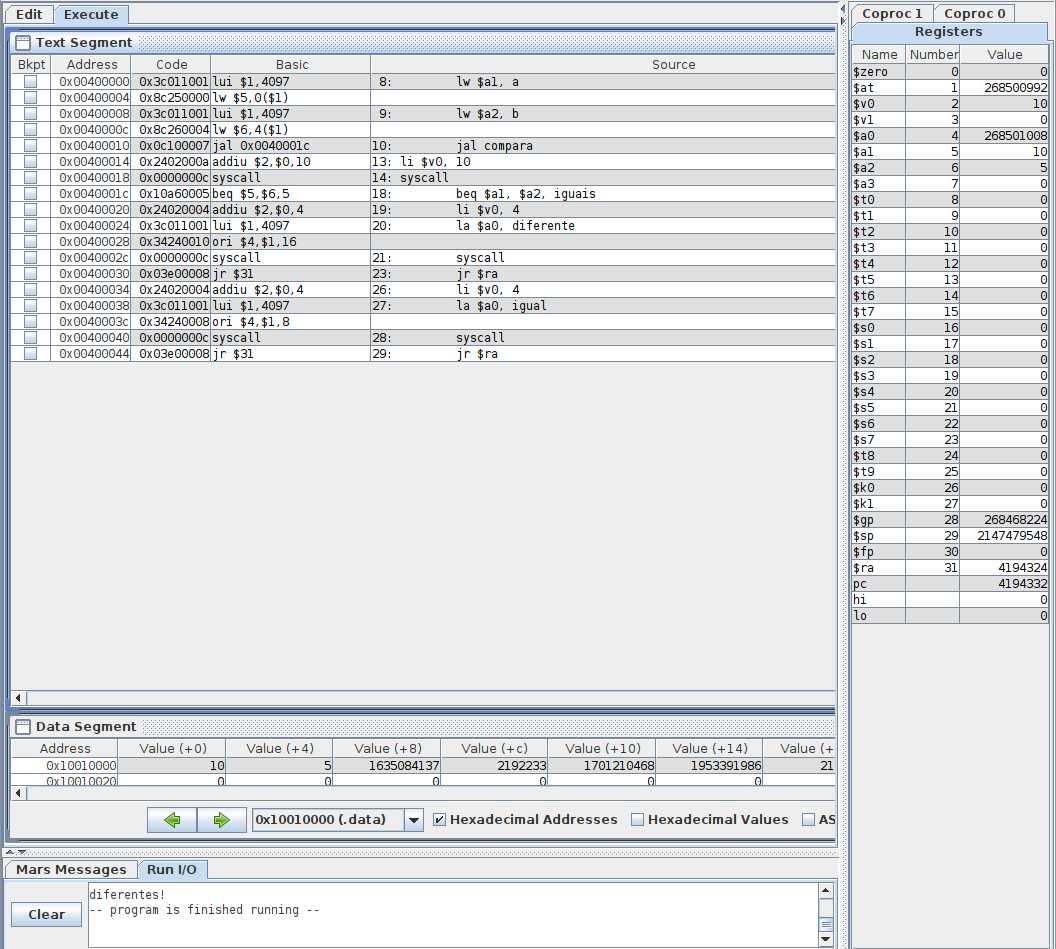
Para cada exercício abaixo, salve o programa no arquivo indicado e cole o print da última tela do simulador.

* + 1. Faça um programa em Assembly que troque o conteúdo de duas variáveis inteiras, **a** e **b**, armazenadas em memória. Salve no arquivo: troca.asm.

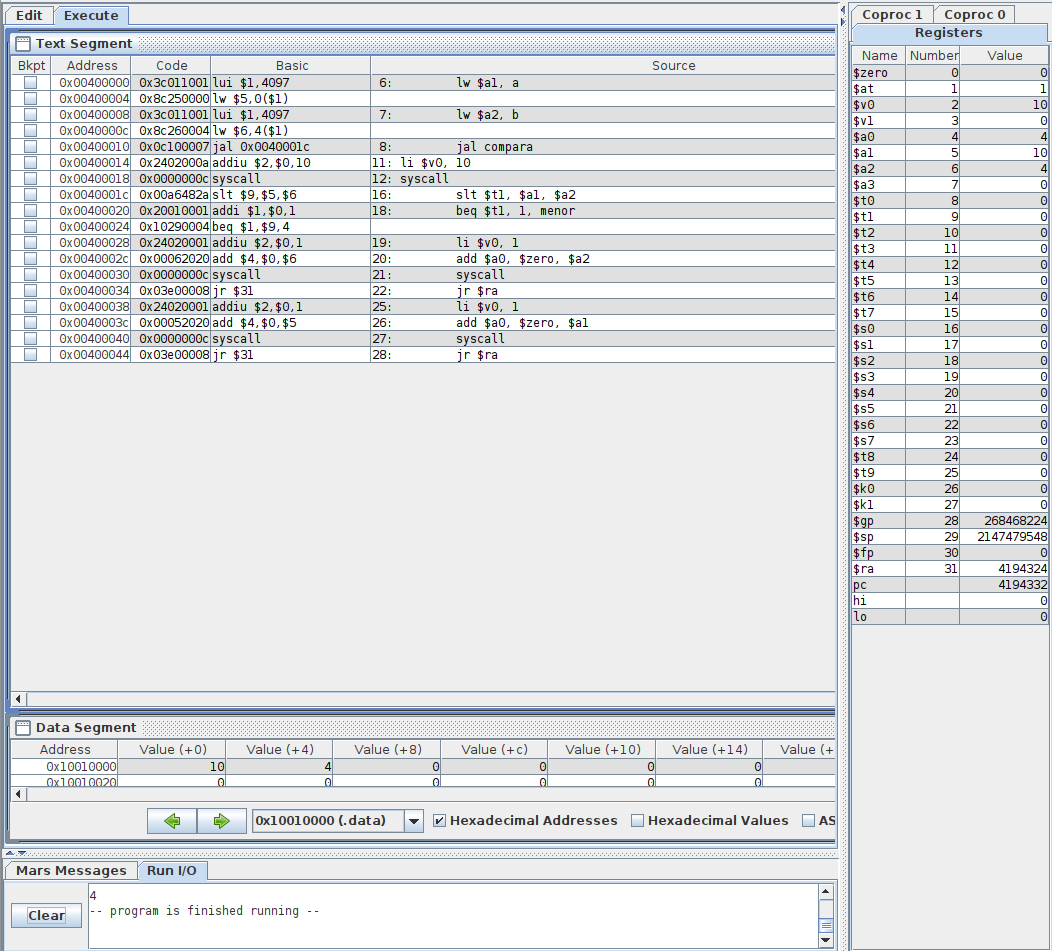
**Antes Depois**

****

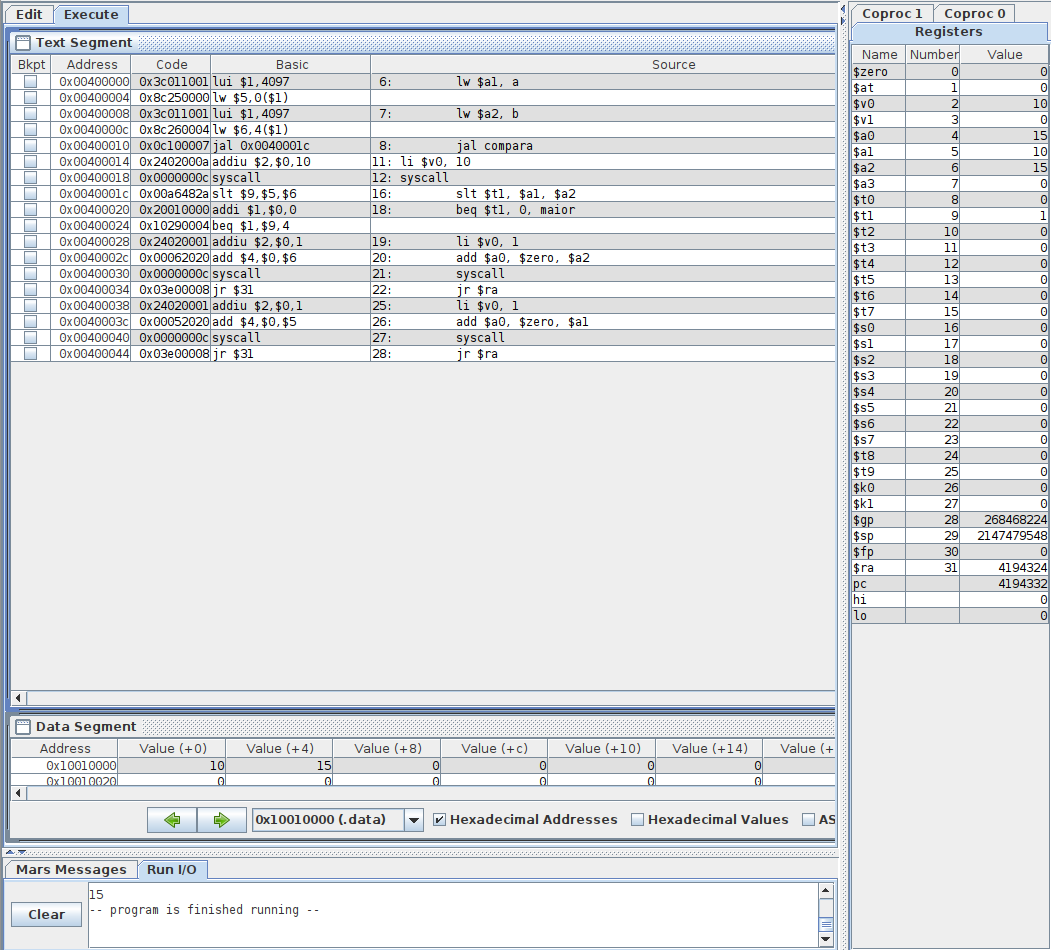
* + 1. Faça um programa que leia duas variáveis inteiras, **a** e **b**, da memóriae imprima “iguais!”, se as variáveis tiverem o mesmo valor; caso contrário, imprima “diferentes!”. Salve no arquivo: compara.asm.

****

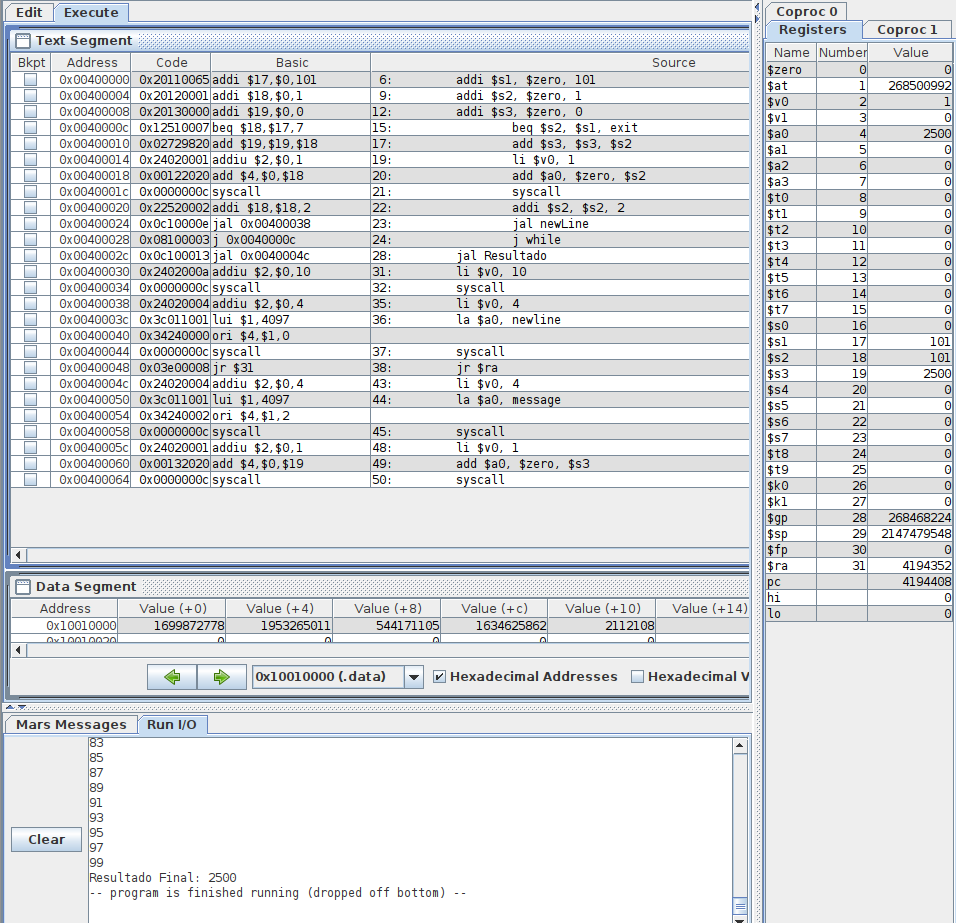
* + 1. Faça um programa que leia duas variáveis inteiras, **a** e **b**, da memóriae imprima o menor valor. Salve no arquivo: menor.asm.



* + 1. Faça um programa que leia duas variáveis inteiras, **a** e **b**, da memóriae imprima o maior valor. Arquivo: Salve no arquivo: maior.asm.

****

* + 1. Faça um programa que imprima o resultado da soma de todos os números ímpares entre 0 e 100. Salve no arquivo: soma\_impares.asm.



* + 1. Faça um programa que leia da memória um número inteiro **n** e imprima o n-ésimo termo da série de Fibonacci. Salve no arquivo: fibonacci.asm.

