



Universidade Federal de Campina Grande  
Unidade Acadêmica de Sistemas e Computação  
Disciplina: *Organização e Arquitetura de Computadores*  
Profa. *Joseana Macêdo Fachine Régis de Araújo*

### Lista de Exercícios Prática 04 (Processador - Assembly)

**Obs.:** Esta atividade deverá ser realizada por uma equipe de 03 (três) alunos.

- SIMULADORES (sugestões)
  - Simulador de Icaro – página de LOAC
  - RISC-V (SPIKE)
    - [Descrição da instalação do simulador SPIKE](#)
    - [Informações adicionais sobre comandos do SPIKE](#)
  - RISC-V (Venus)
    - <https://www.kvakil.me/venus/>
  - RISC-V (Jupiter)
    - <https://github.com/andrescv/Jupiter>
  - INTEL (usando o LINUX)
    - [Linux Assembly Tutorial](#)
    - [Learn Assembly Language](#)
    - [NASM Tutorial](#)
  - Outros
    - [Assembly x86 Emulator](#)
    - [SIMPLE MICROPROCESSOR SIMULATOR](#)

1. Implementar um programa em Assembly (processadores RISC-V, Intel ou similar) que apresente na tela a mensagem "Hello, World!".
2. Implementar o trecho de código abaixo (RISC-V, 32 bits).
  - a) Qual o conteúdo dos registradores ao final da execução de cada instrução?
  - b) Qual o código de máquina de cada instrução?
  - c) Exibir o armazenamento do código de máquina da primeira instrução.

```
addi a0, zero, 5
addi a1, zero, 10
addi a2, zero, 5
add a3, a0, a1
sub a3, a3, a2
```

3. Implementar o trecho de código abaixo (RISC-V, 32 bits).
1. Qual a operação realizada por esse trecho de código?
  2. Qual o conteúdo dos registradores *s0* e *s1* ao final da execução?

```
addi    s0, zero, 1
addi    s1, zero, 0
addi    t0, zero, 128
while:
beq      s0, t0, done
slli    s0, s0, 1
addi    s1, s1, 1
j       while
done:
```

4. Implementar um programa em Assembly (RISC-V, Intel ou similar) que executa o seguinte conjunto de instruções e informar qual o conteúdo final dos registradores envolvidos.

```
g = 3;
h = 2;
i = 5;
j = 1;
f = (g+h) + (i-j);
```