

Atividades de Laboratório 5

Objetivos

O objetivo desta atividade é exercitar o uso de instruções aritméticas e a manipulação de entrada e saída utilizando o conjunto de instruções da arquitetura RISC-V.

Descrição

Neste laboratório, você deve fazer um programa em linguagem de montagem do RISC-V que leia e calcule o valor aproximado da raiz quadrada de números inteiros.

Entrada e saída

Seu programa deve ler da entrada padrão quatro valores em base decimal e imprimir a raiz quadrada de cada um deles.

A entrada e saída serão cadeias de caracteres ASCII, cada uma representando quatro números inteiros de 4 dígitos na base decimal separados por um espaço. Elas devem ser terminadas pelo caracter '\n'.

Exemplo

Entrada	0400 5337 2240 9166
Saída	0020 0073 0047 0095

Observações e Dicas

Linguagem de programação

Esta é uma atividade que deve ser realizada programando-se em linguagem de montagem - A submissão de programas em linguagem de programação de alto nível, como C, ou de programas gerados por ferramentas de compilação, serão consideradas fraude.

Implementação da Raiz Quadrada

- Sugerimos a utilização do [Babylonian method](#) com 10 iterações. Considere que queremos calcular a raiz quadrada de um número y , a ideia básica do método é:

- a. Calcule uma estimativa inicial (*guess*) da raiz:

$$k = \frac{y}{2}.$$

- b. Aproxime sua estimativa, k , do valor real da raiz aplicando a seguinte equação:

$$k' = \frac{k + \frac{y}{k}}{2}$$

- c. Cada aplicação da fórmula acima é considerada “uma iteração”. Para este laboratório, utilize 10 iterações.
- Neste laboratório, aceitaremos soluções aproximadas para o problema.
 - a. Serão consideradas corretas soluções com erro absoluto menor que 10.
 - Você pode utilizar outros métodos para aproximar a raiz quadrada, desde que:
 - a. Utilize apenas inteiros. Não podem ser utilizados números em ponto flutuante ou a instrução de raiz quadrada do RISC-V.
 - b. A aproximação seja tão ou mais precisa quanto a do método sugerido.

Entrada e saída

Para efetuar a leitura e escrita dos dados no terminal, você deve utilizar as chamadas de sistema *read* e *write* (semelhante ao lab 3, mas agora em linguagem de montagem).

Exemplo de *read*:

```
li a0, 0 # file descriptor = 0 (stdin)
la a1, input_adress # buffer
li a2, 1 # size (lendo apenas 1 byte, mas tamanho é variável)
li a7, 63 # syscall read (63)
ecall

input_adress: .skip 0x10 # buffer
```

Exemplo de *write*:

```
li a0, 1 # file descriptor = 1 (stdout)
la a1, string # buffer
li a2, 19 # size
li a7, 64 # syscall write (64)
ecall

string: .asciz "Hello! It works!!!\n"
```