DCC006: Organização de computadores I

Trabalho Prático #1

Professor: Daniel Fernandes Macedo

Antes de começar seu trabalho, leia todas as instruções abaixo.

- O trabalho deve ser feito individualmente. Cópias de trabalho acarretarão em devida penalização `as partes envolvidas.
- Entregas após o prazo serão aceitas, porém haverá uma penalização. Quanto maior o atraso maior a penalização.
- Submeta apenas um arquivo .zip contendo as suas soluções e um arquivo .txt com seu nome e matrícula. Nomeie os arquivos de acordo com a numeração do problema a que se refere. Por exemplo, o arquivo contendo a solução para o problema 1 deve ser nomeado 1.s. Se for solicitado mais de uma implementação para o mesmo problema nomeie 1a.s, 1b.s e assim por diante.
- O objetivo do trabalho ´e praticar as suas habilidades na linguagem assembly. Para isso, você utilizará o *Venus Simulator* (https://www.kvakil.me/venus/). Venus é um simulador de ciclo único que te permite enxergar o valor armazenado em cada registrador e seguir a execução do seu código linha a linha. O simulador foi desenvolvido por Morten Petersen e possui a ISA do RISC-V, embora apresente algumas alterações. Você pode utilizar o seguinte link: https://github.com/mortbopet/Ripes/blob/master/docs/introduction.md para verificar as modificações da sintaxe ISA utilizada pelo simulador. Note que no livro e material da disciplina os registradres são de 64 bits, mas o simulador utilizada registradores de apenas 32 bits. Para utilizar o simulador basta você digitar seu código aba *Editor* e para executá-lo basta utilizar a aba *Simulator*

Problema 1: Sequência 3n+1 (5 pontos)

Implemente um procedimento em *assembly* que calcule o n-ésimo termo da sequência 3n+1. Considere que o número *n* está no registrador x10, e o número de início está no registrador x11. O retorno do procedimento deve acontecer no registrador x10.

A sequência de 3n+1 define um elemento da seguinte forma:

$$A_{n+1} = \begin{cases} \dfrac{A_n}{2}, & se\ A_n\ \text{é}\ par \\ 3 \times A_n + 1, & se\ A_n\ \text{é}\ ímpar \end{cases}$$

A sequência, quando chega ao fim, tem fim quando chegamos ao valor 1. Assim, se temos um valor inicial igual a 11, a sequência seria 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1. Para o valor 11, o 5º termo seria então o valor 26. Caso o valor de n solicitado seja superior ao número de elementos totais da sequência, retorne zero. Por exemplo, caso o programador solicite o vigésimo termo da sequência com valor inicial igual a 11, o seu procedimento deve retornar zero.

Problema 2: Ordenação recursiva (5 pontos)

Implemente um procedimento em *assembly* que faça a ordenação usando o método de seleção. Esta implementação deve ser recursiva. A ordenação por seleção pode ser pensada com o seguinte pseudo-algoritmo:

```
seleção_rec(vetor, nelem)
    se nelem é maior que 2:
        pos ← índice do elemento do vetor com nelem elementos de
        maior valor;
        Troca os elementos pos e nelem-1 de posição;
        Chama seleção_vec(vetor,nelem-1)
    senão
        retorna
```

Considere que o ponteiro para o começo do *array* está em x10 e que o tamanho do *array* está em x11.

Dica: você pode usar o registrador x2 *(stack pointer)* para armazenar os valores dos parâmetros de cada chamada recursiva na memória.

Dicas e sugestões

- Não deixe o trabalho para o último dia. Não viva perigosamente!
- Comente seu código sempre que possível. Isso será visto com bons olhos.