

Trabalho Prático #1

Professor: Omar Paranaíba Vilela Neto e Luiz Filipe Vieira

Monitora: Aline Fernanda Furtado Silva

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

- O trabalho deve ser feito individualmente. Cópias de trabalho acarretarão em devida penalização às 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 é 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/wiki/RISC-V-Assembly-Programmer%27s-Manual-\(Adapted-for-Ripes\)](https://github.com/mortbopet/Ripes/wiki/RISC-V-Assembly-Programmer%27s-Manual-(Adapted-for-Ripes)) para verificar as modificações da sintaxe ISA utilizada pelo simulador. Note que no livro e material da disciplina os registradores são de 64 *bits*, mas o simulador utiliza registradores de apenas 32 *bits*. Para utilizar o simulador basta você digitar seu código aba **Editor** e para executá-lo basta utilizar a aba **Simulador**.

Problema 1: Potência Elétrica

(2 pontos)

Potência elétrica é a medida da quantidade de energia elétrica fornecida ou consumida por um circuito elétrico. Pode ser calculada por meio de grandezas como tensão elétrica (V), resistência elétrica (R) e corrente elétrica (I), se relacionam através, por exemplo da seguinte equação:

$$P = V \times I$$

Escreva um programa *assembly* (RISC-V) que, dadas duas dessas grandezas quaisquer, o programa possa calcular e retornar o valor da terceira grandeza. Considere `x10`, `x11` e `x12` como potência (P), tensão (V) e (I) respectivamente e a variável com valor zero é a que deve ser calculada. Caso mais de uma variável tenha valor zero, seu programa deve retornar também zero.

Problema 2: Fatorial

(3 pontos)

Considere o seguinte código em C para uma primeira aproximação a uma função que calcula iterativamente o fatorial de um número.

```
int fact (int n) {
    int i, res;
    res = 1;
    for (i = 2; i <= n; i++)
        res = res * i;
    return (res);
}
```

Implemente o algoritmo acima em *assembly* (RISC-V), considere que o número n está no registrador `x10`.

Problema 3: Ordenação por seleção

(5 pontos)

Se você pedir para alguém que não é familiarizado com algoritmos para que ordene uma sequência de números, há uma probabilidade alta dessa pessoa aplicar o Selection Sort. Isso porque esse algoritmo tem uma rotina básica muito simples e direta: selecionar (daí o nome Selection Sort) o menor elemento da sequência e colocar esse elemento na primeira posição do array.

A ideia é executar várias vezes esses dois passos para ordenar um array. Para ser exato, se executarmos N vezes esses dois passos em um array, controlando os índices em que os passos são executados, o resultado é a ordenação completa do mesmo.

Abaixo uma possível implementação do *selection sort* em C:

```
int indice_menor = 0;
for (int i=1; i < v.length; i++) {
    if (v[i] < v[indice_menor])
        índice_menor = i;
}
// coloca o menor na primeira posição
Int aux = v[0];
V[0] = v[índice_menor];
V[índice_menor] = aux;
}
```

Implemente o *selection sort* em *assembly (RISC-V)*, considere que o começo do array está em x10 e o tamanho do array está em x11. Para esta atividade você pode utilizar arrays ou ponteiros.

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.
- A monitora está disponível para atender a quaisquer dúvidas sobre esse trabalho. Segue o email para contato com a monitora: furtadoaline@dcc.ufmg.br Insira a tag [DCC006] no assunto do email.