

Linguagens de Programação 2019/2020

Departamento de Informática, Universidade de Évora

2º Trabalho Prático

– Execução de programas TISC –

1 Objectivo

Utilizando a linguagem de programação **Java** (ou outra à escolha), pretende-se implementar a máquina **TISC**.

Neste segundo trabalho, a implementação deverá ler um programa TISC na sua entrada padrão, executá-lo e terminar quando o programa terminar, tendo por base o carregador de programas TISC desenvolvido no 1º trabalho.

2 Características de Implementação

A descrição da máquina TISC, a linguagem iPascal e as instruções TISC estão descritas no enunciado do 1º trabalho.

2.1 Memória da máquina TISC

A memória de instruções foi desenhada no 1º trabalho. A pilha de avaliação será uma pilha de inteiros e a memória de execução será análoga a um vetor de inteiros.

Poderão existir outras estruturas de dados somente para o controlo do funcionamento da máquina. Estas estruturas não poderão ser usadas durante a execução do programa TISC para manipular os dados com que este trabalha.

2.2 Registos de ativação

Faz parte da realização do trabalho o desenho dos registos de ativação.

2.3 Entrada e saída de dados

A implementação da máquina TISC lerá o programa da sua entrada padrão e não escreverá nada no ecrã, excepto o resultado das instruções de saída incluídas no programa TISC em execução, que deverá ser enviado para a saída padrão da máquina (**System.out**).

2.4 Exemplos

Deverá utilizar os exemplos de programas TISC fornecidos no 1º trabalho para testar a implementação da máquina.

3 Relatório

O relatório a entregar deverá incluir:

- O desenho adoptado para os registos de ativação e a sua justificação
- A descrição da estrutura da implementação da máquina e das estruturas de dados usadas
- A descrição, através de pseudo-código, do funcionamento de cada instrução TISC, tal como implementada no trabalho. Por exemplo, o pseudo-código para a instrução **sub** poderia ser:

```
sub:
    o2 = desempilha()
    o1 = desempilha()
    empilha(o1 - o2)
    PC = PC + 1
```

(considerando que empilha e desempilha são duas primitivas da pilha de avaliação).

Para a instrução **salta**, poderíamos ter:

```
salta etiqueta:
    PC = @etiqueta
```

onde **@** é um operador unário que devolve o endereço do operando.

4 Entrega e Avaliação

O trabalho será realizado por grupos de **dois alunos**. Os elementos a entregar são:

- os ficheiros com o código fonte da implementação da máquina, incluindo os ficheiros com as definições dos analisadores lexical e sintático
- um Makefile com os comandos para a construção da implementação da máquina
- o relatório

Estes elementos deverão ser entregues através do *moodle* num ficheiro de nome **xxx_xxx.zip** (ou **xxx_xxx.tar.gz**), onde **xxx** é o número de cada um dos alunos que fazem parte do grupo. A data de entrega estará disponível no *moodle*.

O trabalho realizado por cada grupo será apresentado em data a combinar. Apesar do trabalho ser de grupo, cada aluno, a título individual, tem a responsabilidade de responder por todo o trabalho. Assim, é indispensável que cada membro do grupo programe efectivamente.

O trabalho deverá ser realizado somente por quem o entrega. A detecção de qualquer tentativa de subverter este princípio terá como consequência, para todos os envolvidos, a impossibilidade de obter aprovação à disciplina este ano lectivo.

Agradece-se a comunicação de qualquer erro detetado no enunciado do trabalho.

Bom trabalho!