Linguagens de Programação

Leitura de Programas
TICS

Ano Lectivo

2019/2020

—

Fernando Prates nº 34197

Ana Silvério nº 37561

Descrição da máquina TISC

Máquina TISC e Zonas de Memória

Uma máquina **TISC**, *Tiny Instruction Set Computer*, é uma máquina abstrata simples que está preparada para a execução de programas escritos em linguagens estruturadas em blocos, com ambientes de âmbito estático e definição de funções e procedimentos locais.

Estas máquinas têm três zonas de memória distintas:

- **Memória de Instruções**: Zona onde são armazenadas as instruções do programa a executar, num formato não especificado.

- **Pilha de Avaliação**: Zona usada na avaliação de expressões, para a transferência de dados e para guardar o valor de retorno de uma função.

- **Memória de Execução**: Zona onde se encontram os registos de ativação dos blocos do programa cuja execução ainda não terminou.

Máquina TISC e Registos

As máquinas TISC, além de terem as zonas particularmente especificadas de memória, são possuidoras de dois registos muito importantes:

- **Registo EP**: Registo que contém o *environment pointer*.

- **Registo PC**: Registo que remete, direcciona, para o *program counter*.

Implementação

Descrição dos Analisadores: Lexical e Sintático

A leitura de programas TISC foi desenvolvida através da linguagem de programação JAVA. Como tal, foram utilizados para o desenvolvimento de analisadores lexical e sintático os geradores JLEX e CUP.
Este são geradores de analisadores lexicais e analisadores sintáticos próprios para a linguagem de programação JAVA, respetivamente.

O conteúdo do ficheiro TISC.cup foi alterado de modo a que este possa interagir funcionalmente com o ficheiro TISC.lex, sem que ocorram falhas ou excepções, e que seja possivel a introdução de instruções na máquina TISC.

Descrição do Registo de Ativação

Implementado na Classe RegistoAtivacao. Nesta Classe encontram-se declarados os seguintes constituintes do Registo de Ativação:

- **ResgistoAtivacao *ControlLink***: Apontador para a função que chamou e gerou o registo de ativação.

- **EnderecoRetorno**: Variavel do tipo int, que representa o *program counter* da posição a partir de onde o programa deve retomar após a conclusão da função.

- **args**: Arrays de argumentos, do tipo int, do registo de ativação, int [ ] args.

- **vars**: Arrays de variaveis, do tipo int, do registo de ativação, int [ ] vars.

- **nargs**: Números de argumentos que a função vai receber, int nargs.

- **nvars**: Número de variaveis que a função vai receber, int nvars.

Implementação

Descrição da Máquina TISC: Classe TISC

- **ArrayList<Instrucao> ins_memory**: *Array list* utilizado para guarder todas as instruções lidas e são organizadas por ordem de leitura de forma a facilitar a sua utilização. Ou seja, a primeira instrução a ser lida é a primeira a ser chamada.

- **Stack<Integer> aval_pilha**: Pilha de valores do tipo int, que tem como finalidade fazer ou auxiliar nos cálculos e operações da máquina.

- **Stack<RegistoAtivacao> exec_memory**: *Stack* que contém os registos de ativação de modo poder-se proceder à organização dos blocos constituintes do programa.

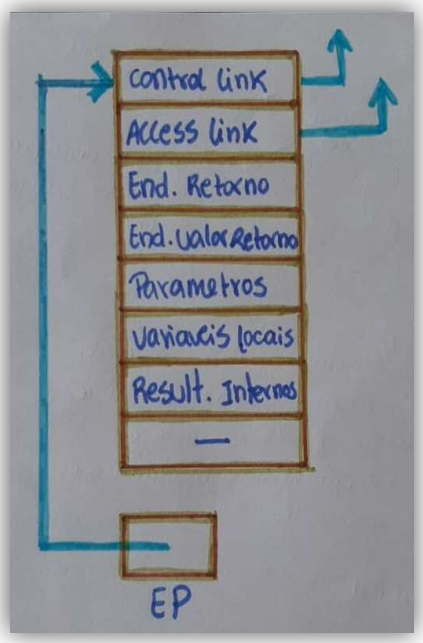
- **Hashtable<String, Integer> etiquetas**: *Hash Table* de objectos etiquetas. Esta *hash table* tem como chave, *key*, o nome da etiqueta, função do programa. O objecto etiqueta é que contém a posição onde esta se encontra. Quer isto dizer, os objectos etiquetas estão guardados numa *hash table* em que o nome da função é a chave, *key*, e as etiquetas são o valor correspondente.

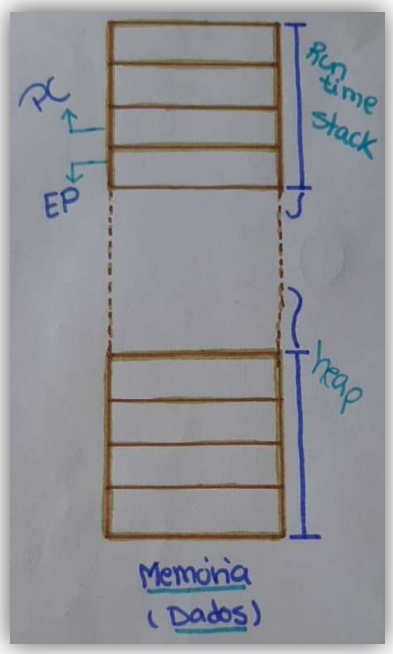
- **RegistoAtivacao EP**: Apontador representativo do apontador de ambiente, environment pointer, que aponta para o registo de ativação corrente.

- **int PC**: Variável, do tipo int, representativa do program counter. Indica qual a instrução onde se encontra a máquina.

Implementação

Esquemas para a percepção do Registo de Ativação & PC e EP





Descrição das Instruções

As instruções são objetos que respeitam uma hierarquia. A Classe Instrucao é a classe principal da máquina. Cada instrução pertence à sua própria classe e esta é uma extenção da classe Instrução. Ou seja, os diferentes tipos de instruções que existem na máquina TISC são extendidos da Classe Instrucao. Então a Classe Instrucao será uma Superclasse e as classes dos diferentes tipos intruções da máquina TISC são Subclasses. O mesmo irá ocorrer para cada instrução. As classes de um tipo de instrução são Superclasses e as instruções em específico desse tipo são extendidas da classe do tipo de instrução, Subclasses.

As instruções são lidas e colocadas em memória no ArrayList ins_mem, que contém todas as instruções do programa lido por ordem de leitura, mencionado anteriormente.