UNISC
UNIVERSIDADE DE SANTA CRUZ DO SUL

Computabilidade – Trabalho 1 – Simulador máquina Norma

NOME: Gabriel Lopes, Henyo Nunes e Lucas Rodrigues

DATA: 12/12/2022

Link GitHub: RodriguesLucas/maquinaNorma (github.com)

Tal como a máquina norma, nosso simulador possibilita:

- **Teste:** x_zero: Testa se o registrador x vale zero.
- **Operações:** o ad_x: Incrementa o registrador x.
- **Operações:** o sub_x: Decrementa o registrador x.

Descrição do funcionamento:

- O algoritmo lê as instruções
- Faz a separação dos rótulos, comandos e valores.
- realiza as operações, testagem, adição e subtração
- Ao chegar ao fim da instrução, realiza sua parada

.

Estrutura de dados utilizadas:

Neste simulador, utilizamos a estrutura de lista, e utilizamos a linguagem Java.

Lógica do funcionamento:

• O algoritmo lê as instruções por meio de um documento, já preenchido previamente pelo usuário.

• Faz a separação dos rótulos, comandos e valores, por meio da comparação do comando, salvando, após um número x de casas, o valor ou o comando, para ser utilizado posteriormente para o funcionamento do algoritmo.

• Por meio dos dados separados, realiza as operações, testagem, adição e subtração

• Possibilita a utilização de macros pré-definidos para realização de algumas operações, definidas pelo usuário.

• Ao chegar ao fim da instrução, realiza sua parada.

A lógica do programa foi separada basicamente em cinco funções:

getRegistradores() - Responsável por criar uma lista de objeto (Registrador), com todos os registradores definidos pelo usuário no arquivo.

getInstrucoes() - Responsável por pegar todas as instruções do arquivo, transformando-a em um objeto (Rótulo), que posteriormente será utilizado na lógica de execução.

getMacros() - Método que percorre todos os arquivos de macros criando uma lista com objeto (Macro).

executa() - Percorre a lista de instruções (Rótulos), fazendo as operações correspondentes a cada posição da lista. Tendo a parada definida pelo usuário no código de execução.

executaMacro() - Percorre a lista de Macros criadas anteriormente, e chama o método **executa().**

Prints das Telas:

Sem Macro

```
Computação:
(1,(0,5))
(2,(0,5))
(3,(1,5))
(1,(1,4))
(2,(1,4))
(3,(2,4))
(1,(2,3))
(2,(2,3))
(3,(3,3))
(1,(3,2))
(2,(3,2))
(3,(4,2))
(1,(4,1))
(2,(4,1))
(3,(5,1))
(1,(5,0))
Valor Registradores:
[id= a, value= 5, id= b, value= 0]
```

Com Macro

```
Computação:
(1,(0,3))
  Início de Macro
  (1,(0,3,0))
  (2,(0,3,0))
  (3,(1,3,0))
  (4,(1,3,1))
  (1,(1,2,1))
  (2,(1,2,1))
  (3,(2,2,1))
  (4,(2,2,2))
  (1,(2,1,2))
  (2,(2,1,2))
  (3,(3,1,2))
  (4,(3,1,3))
  (1,(3,0,3))
  (5,(3,0,3))
  (6,(3,0,3))
  (7,(3,1,3))
  (5,(3,1,2))
  (6,(3,1,2))
  (7,(3,2,2))
  (5,(3,2,1))
  (6,(3,2,1))
  (7,(3,3,1))
  (5,(3,3,0))
  Fim de Macro
(3,(3,3))
(1,(3,2))
    Início de Macro
    (1,(3,2,0))
    (2,(3,2,0))
    (3,(4,2,0))
    (4,(4,2,1))
    (1,(4,1,1))
    (2,(4,1,1))
    (3,(5,1,1))
    (4,(5,1,2))
    (1,(5,0,2))
    (5,(5,0,2))
    (6,(5,0,2))
    (7,(5,1,2))
    (5,(5,1,1))
    (6,(5,1,1))
    (7,(5,2,1))
    (5,(5,2,0))
    Fim de Macro
  (3,(5,2))
  (1,(5,1))
    Início de Macro
    (1,(5,1,0))
    (2,(5,1,0))
    (3,(6,1,0))
    (4,(6,1,1))
    (1,(6,0,1))
    (5,(6,0,1))
    (6,(6,0,1))
    (7,(6,1,1))
    (5,(6,1,0))
    Fim de Macro
  (3,(6,1))
  (1,(6,0))
  Valor Registradores:
  [id= a, value= 6, id= b, value= 0]
```