

## Elementos de Sistemas - Projeto j - Ferramental - VMTranslator

Rafael Corsi - rafael.corsi@insper.edu.br

Maio, 2018

TODO de inicialização - Scrum Master:

1. Atualizar arquivo ScramMaster.txt
2. Atualizar repositório com o upstram
3. Criar projeto no github
4. Atribuir tarefas e acompanhar o desenvolvimento

## VM Translator

O VMTranslator é um programa escrito em Java que faz a tradução de códigos escrito na linguagem VM definida no curso e traduz para linguagem Assembly do computador Z01.

## Testando

```
scripts/testeVmMytraslator.py
```

```

VMTranslator      Assembler      \  Z01-Simulator  -----
arquivo.vm -----> .nasm -----> .hack > -----> - Verifica saída -
                                     ^
                                     |
                                     |- Desenvolvido no projeto H
```

1. Traduzir o arquivo .vm -> .nasm

Para isso foi criado alguns programas (I-VM/src/vmExamples/) em VM muito específicos que testão somente um comando, ou uma parte da tradução do VM-

Translator. Por exemplo o teste SimpleAdd possui somente a seguinte linha :

**add**

Esse teste foi criado para testar o Code.writeArithmetic no caso de um comando add. Para isso, antes da execução desse código, o simulador faz a inicialização da RAM, simulando valores na pilha e já configurando o SP para uma situação real. A memória antes da execução da instrução add é a seguinte :

```
0 : 0000000100000010;
256 : 0000000000000010;
257 : 0000000000000100;
258 : 0000000000000000;
```

(J-VMtranslator/tests/tst/SimpleAdd/SimpleAdd0\_in.mif)

Espera-se o resultado final após a execução do comando add :

```
0 : 0000000100000001
256 : 0000000000000110
```

(J-VMtranslator/tests/tst/SimpleAdd/SimpleAdd0\_tst.mif)

## **A seguir lista dos testes e suas implementações :**

### **code.writePushPop**

- pop
  - SimplePopTemp : pop temp ...
  - SimplePopLocal : pop local ...
  - SimplePopThat : pop that ...
  - SimplePopThis : pop this ...
- push
  - SimplePushConst : push constant ....
  - SimplePushTemp : push tempo ....
  - SimplePushLocal : push local ....
  - SimplePushArg : push argument ...
  - SimplePushThis : push this ...
  - SimplePushThat : push that ...

### **code.writeArithmetic**

- SimpleAdd : add
- SimpleNeg : neg
- SimpleSub : sub
- SimpleEq : eq

- SimpleGt : gt
- SimpleLt : lt
- SimpleAnd : and
- SimpleOr : or

! Para esses testes funcionarem, o Parser deve estar pronto e passando nos testes anitários.