



**Universidade Estadual de Santa Cruz – UESC**

## **Relatório p-code Machine**

**Discente: Levy Santiago**

**Matrícula: 201520138**

**Disciplina Compiladores.**

**Curso Ciência da Computação**

**Semestre 2018.2**

**Professor César Bravo**

**Ilhéus – BA**

**2018**

# Índice

|   |          |
|---|----------|
| <b>Tabela de Operações</b>                          | <b>2</b> |
| <b>Compilando o projeto</b>                         | <b>3</b> |
| <b>Executando o projeto</b>                         | <b>3</b> |
| <b>Testes realizados</b>                            | <b>3</b> |
| Contagem de 1 até 100                               | 3        |
| Calcular a expressão $2^2 + 3 \cdot 2 + 1$          | 4        |
| Calcular $1+2+3+4+5+6+7+8+9+10$                     | 4        |
| Calcular $1^2+2^2+3^2+4^2+5^2+6^2+7^2+8^2+9^2+10^2$ | 5        |
| <b>Link para todo o conteúdo</b>                    | <b>6</b> |

## Tabela de Operações

A seguinte codificação de operações foi considerada:

| Código | Símbolo | Semântica   |
|--------|---------|---|
| 0      | Return  | Retorna o controle à rotina que fez a chamada           |
| 1      | neg     | $x = \text{pop}(); \text{push}(-x)$                     |
| 2      | +       | $x = \text{pop}(); y = \text{pop}(); \text{push}(y+x)$  |
| 3      | -       | $x = \text{pop}(); y = \text{pop}(); \text{push}(y-x)$  |
| 4      | *       | $x = \text{pop}(); y = \text{pop}(); \text{push}(y*x)$  |
| 5      | Div     | $x = \text{pop}(); y = \text{pop}(); \text{push}(y/x)$  |
| 6      | odd     | TRUE (iff) $s[t]$ is odd                                |
| 7      | ==      | $x = \text{pop}(); y = \text{pop}(); \text{push}(y==x)$ |
| 8      | <>      | $x = \text{pop}(); y = \text{pop}(); \text{push}(y<>x)$ |
| 9      | <       | $x = \text{pop}(); y = \text{pop}(); \text{push}(y<x)$  |
| 10     | >=      | $x = \text{pop}(); y = \text{pop}(); \text{push}(y>=x)$ |
| 11     | >       | $x = \text{pop}(); y = \text{pop}(); \text{push}(y>x)$  |
| 12     | <=      | $x = \text{pop}(); y = \text{pop}(); \text{push}(y<=x)$ |

## Compilando o projeto

Para compilar o projeto basta executar o comando:

```
$ gcc main.c -o machine -Wall
```

## Executando o projeto

O projeto pode ser executado recebendo um arquivo de entrada ou inserindo o código (p-code) diretamente no terminal.

Para executar o projeto usando um arquivo como entrada basta inserir como argumento o caminho para o arquivo que contém o p-code, por exemplo:

```
$ ./machine examples/pow1to10
```

Para executar diretamente pelo terminal, basta executar o código normalmente:

```
$ ./machine
```

## Testes realizados

Abaixo são apresentados alguns códigos (p-code) que foram utilizados para testar o projeto.

### Contagem de 1 até 100

```
INT 0 2  
LIT 0 1  
STO 0 0  
LIT 0 1  
STO 0 1  
LOD 0 0  
LOD 0 1  
OPR 0 2  
STO 0 1  
LOD 0 1  
LIT 0 100  
OPR 0 7  
JPC 0 5
```

### Calcular a expressão $2^2 + 3 \cdot 2 + 1$

```
INT 0 4
LIT 0 1
STO 0 0
LIT 0 2
STO 0 1
LIT 0 3
STO 0 2
LOD 0 1
LOD 0 1
OPR 0 4
LOD 0 2
LOD 0 1
OPR 0 4
LOD 0 0
OPR 0 2
OPR 0 2
```

Calcular  $1+2+3+4+5+6+7+8+9+10$

```
INT 0 3
LIT 0 1
STO 0 0
LIT 0 2
STO 0 1
LOD 0 0
LOD 0 1
OPR 0 2
STO 0 2
LOD 0 0
LOD 0 1
OPR 0 2
STO 0 1
LOD 0 1
LOD 0 2
OPR 0 2
STO 0 2
LOD 0 1
LIT 0 10
OPR 0 7
JPC 0 9
```

Calcular  $1^2+2^2+3^2+4^2+5^2+6^2+7^2+8^2+9^2+10^2$

```
INT 0 3
LIT 0 1
STO 0 0
LIT 0 2
STO 0 1
LOD 0 1
LOD 0 1
OPR 0 4
LOD 0 0
OPR 0 2
STO 0 2
LOD 0 0
LOD 0 1
OPR 0 2
STO 0 1
LOD 0 1
LOD 0 1
OPR 0 4
LOD 0 2
OPR 0 2
STO 0 2
LOD 0 1
LIT 0 10
OPR 0 7
JPC 0 11
```

Link para todo o conteúdo

<https://github.com/Levysantiago/p-code-machine>