



**Universidade Estadual de Santa Cruz – UESC**

## **Relatório Projeto 1D**

**Discente: Levy Santiago**

**Matrícula: 201520138**

**Disciplina Compiladores.**

**Curso Ciência da Computação**

**Semestre 2018.2**

**Professor César Bravo**

**Ilhéus – BA**

**2018**

## Índice

<b>Código usado</b>	<b>3</b>
<b>Compilando o projeto</b>	<b>3</b>
<b>Executando o projeto</b>	<b>3</b>
<b>Teste realizado</b>	<b>3</b>
Implementação do seguinte código fonte	4
<b>Link para o conteúdo</b>	<b>8</b>

## Código usado

Foi utilizado como base o código p-code machine em c++ que pode ser encontrado em: [cpp.sh/9nsyz](https://cpp.sh/9nsyz). Neste código a implementação do p-code é inserida diretamente do código, na função main. Portanto, o teste realizado deste projeto 1D será representado pela implementação realizada com este código.

## Compilando o projeto

Para compilar o projeto basta executar o comando:

```
$ g++ p-code.cpp -o machine -Wall
```

## Executando o projeto

Para executar o projeto basta executar o seguinte comando com o nome do programa gerado, por exemplo:

```
$ ./machine
```

## Teste realizado

Abaixo é apresentado o código (p-code) que foi implementado, baseado em um código fonte dado pelo professor, para testar na máquina. Foram usados dois vetores e um inteiro somente para ajudar no processo de desenvolvimento do p-code. O inteiro foi utilizado como um contador para o índice do vetor “code”, para não precisar escrever o número da linha de cada instrução. Os dois vetores foram usados somente para facilitar a definição do conteúdo das matrizes, ficando mais fácil alterar os valores de uma matriz em uma linha só do que procurar as linhas na p-code em que o conteúdo da matriz é definido.

## Implementação do seguinte código fonte

mult, soma	main
<pre> mult(A,ar,ac, B,br,bc, C) {   local i, j, k, s;   for(i=0;i&lt;ar;i++){     for(j=0;j&lt;ac;j++){s=0;       for(j=0;j&lt;ac;j++){         S+=A[i][k]*B[k][j];}       C[i][j]=s;}} }  soma(A,ar,ac, B,br,bc, C){   local i, j, ;   for(i=0;i&lt;ar;i++){     for(j=0;j&lt;ac;j++){       A[i][j]=B[i][j]+C[i][j];}} } </pre>	<pre> main() {   local A[2][3];   local B[3][2];   local C[2][2];   local D[2][2];    C = A*B;   D = (C*C) + C; } </pre>

*/\*Um contador para determinar o próximo índice do vetor "code"  
(próxima linha da implementação p-code)\*/*

**int i = 0;**

*/\*Vetores usados para definir o conteúdo das matrizes,  
usado somente no momento do preenchimento das matrizes.\*/*

**int A[6] = {0,0,0,1,1,1};**

**int B[6] = {0,1,0,1,0,1};**

*// Aqui vc preenche as instrucoes no vetor code  
/\*Preenchendo matriz A\*/*

```

code[ i].f = INT; code[ i].l = 0; code[ i++].a = 23;
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = A[0];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 3;
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = A[1];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 4;
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = A[2];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 5;
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = A[3];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 6;
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = A[4];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 7;
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = A[5];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 8;

```

```

/*Preenchendo matriz B*/
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = B[0];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 9;
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = B[1];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 10;
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = B[2];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 11;
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = B[3];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 12;
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = B[4];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 13;
code[ i].f = LIT; code[ i].l = 0; code[ i++].a = B[5];
code[ i].f = STO; code[ i].l = 0; code[ i++].a = 14;

code[ i].f = CAL; code[ i].l = 0; code[ i++].a = 29; // Chamando
função de multiplicação para C = A * B
code[ i].f = CAL; code[ i].l = 0; code[ i++].a = 79; // Chamando
função de multiplicação para D = (C * C)
code[ i].f = CAL; code[ i].l = 0; code[ i++].a = 113; // Chamando
função de soma para D = D + C
code[ i].f = JMP; code[ i].l = 0; code[ i++].a = 131; // Pulando para
o retorno final

/*Iniciando multiplicação de A com B e armazenando em C*/
code[ i].f = INT; code[ i].l = 0; code[ i++].a = 3; // linha 29
code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 3;
code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 9;
code[ i].f = OPR; code[ i].l = 1; code[ i++].a = 4;
code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 4;
code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 11;
code[ i].f = OPR; code[ i].l = 1; code[ i++].a = 4;
code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 5;
code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 13;
code[ i].f = OPR; code[ i].l = 1; code[ i++].a = 4;
code[ i].f = OPR; code[ i].l = 1; code[ i++].a = 2;
code[ i].f = OPR; code[ i].l = 1; code[ i++].a = 2;
code[ i].f = STO; code[ i].l = 1; code[ i++].a = 15;

code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 3;
code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 10;
code[ i].f = OPR; code[ i].l = 1; code[ i++].a = 4;
code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 4;
code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 12;
code[ i].f = OPR; code[ i].l = 1; code[ i++].a = 4;

```

```
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 5;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 14;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;
code[ i ].f = STO; code[ i ].l = 1; code[ i++ ].a = 16;
```

```
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 6;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 9;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 7;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 11;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 8;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 13;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;
code[ i ].f = STO; code[ i ].l = 1; code[ i++ ].a = 17;
```

```
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 6;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 10;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 7;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 12;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 8;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 14;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;
code[ i ].f = STO; code[ i ].l = 1; code[ i++ ].a = 18;
```

```
/*Retorna para a main*/
```

```
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 0;
```

```
/*Iniciando cálculo de C * C e armazenando em D*/
```

```
code[ i ].f = INT; code[ i ].l = 1; code[ i++ ].a = 3; // linha 79
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 15;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 15;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 16;
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 17;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;
```

```
code[ i ].f = STO; code[ i ].l = 1; code[ i++ ].a = 19;
```

```
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 15;  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 16;  
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 16;  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 18;  
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;  
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;  
code[ i ].f = STO; code[ i ].l = 1; code[ i++ ].a = 20;
```

```
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 17;  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 15;  
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 18;  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 17;  
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;  
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;  
code[ i ].f = STO; code[ i ].l = 1; code[ i++ ].a = 21;
```

```
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 17;  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 16;  
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 18;  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 18;  
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 4;  
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;  
code[ i ].f = STO; code[ i ].l = 1; code[ i++ ].a = 22;
```

```
/*Retorna para a main*/
```

```
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 0;
```

```
/*Realizando a soma de D (C * C) com C e armazenando em D*/
```

```
code[ i ].f = INT; code[ i ].l = 1; code[ i++ ].a = 3; // linha 113  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 15;  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 19;  
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;  
code[ i ].f = STO; code[ i ].l = 1; code[ i++ ].a = 19;
```

```
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 16;  
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 20;  
code[ i ].f = OPR; code[ i ].l = 1; code[ i++ ].a = 2;  
code[ i ].f = STO; code[ i ].l = 1; code[ i++ ].a = 20;
```

```
code[ i ].f = LOD; code[ i ].l = 1; code[ i++ ].a = 17;
```

```
code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 21;
code[ i].f = OPR; code[ i].l = 1; code[ i++].a = 2;
code[ i].f = STO; code[ i].l = 1; code[ i++].a = 21;

code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 18;
code[ i].f = LOD; code[ i].l = 1; code[ i++].a = 22;
code[ i].f = OPR; code[ i].l = 1; code[ i++].a = 2;
code[ i].f = STO; code[ i].l = 1; code[ i++].a = 22;

/*Retorna para a main*/
code[ i].f = OPR; code[ i].l = 1; code[ i++].a = 0;

/*RETORNO FINAL*/
code[ i].f = OPR; code[ i].l = 0; code[ i++].a = 0; // linha 131
```

## Link para o conteúdo

Google Drive: <https://drive.google.com/open?id=1jjxFjgJs3aJhz-YBkRwSIheFCPCQW24H>