Universidade Estadual de Santa Cruz – UESC

Relatório p-code machine

Relatório feito por Darley Souza Sampaio

Disciplina Compiladores

Curso Ciência da Computação

Semestre 2022.2

Professor Cesar Alberto Bravo Pariente

Ilhéus – BA
2022

ÍNDICE

Configurações da Máquina

Para fins de conhecimento da máquina utilizada, segue as seguintes configurações onde o código foi transcrito e compilado:

Processador: AMD Ryzen 3200G

Placa de Vídeo: Geforce GTX 1050TI 4GB OC

Placa Mãe: Asrock A320M-HD

Memória: 16GB (2x8) DDR4 2666mHz

Armazenamento: SSD M.2 120GB + HD 1TB

Sistema Operacional: Linux(Mint)

Comandos de Operação

Comandos para realizar as operações.

Comando	Ação
LIT 0 a	armazena valor em a no topo da pilha
OPR 0 a	executa uma operação selecionada por a
LOD l a	carrega a variável a de l
STO l a	salva a variável a em l
CAL l a	chama o procedimento de nível l em a
INT 0 a	adiciona ao registro um tamanho a
JMP 0 a	pula para a
JPC 0 a	pulo condicional para a

Códigos da Operação OPR

Códigos utilizados dentro da operação OPR para realizar as ações.

Código	Símbolo	Ação
0	Return	Return from a subroutine
1	Negate	Negate the value at the top of the stack
2	ADD	x = pop(); y = pop(); push(y + x)
3	Subtract	x = pop(); y = pop(); push(y - x)
4	Multiply	x = pop(); y = pop(); push(y * x)
5	Divide	x = pop(); y = pop(); push(y / x)
6	ODD?	Checks the top value of the stack whether it is odd or not
7	==	x = pop(); y = pop(); push(y == x)
8	< >	x = pop(); y = pop(); push(y < > x)
9	<	x = pop(); y = pop(); push(y < x)
10	>=	x = pop(); y = pop(); push(y >= x)
11	>	x = pop(); y = pop(); push(y > x)
12	<=	x = pop(); y = pop(); push(y <= x)

Compilando o Programa

Para realizar a compilação do programa basta inserir em qualquer compilador. Caso esteja utilizando o linux, utilize os seguintes comandos em seu terminal.

O link para download de todo o projeto em sí é:
https://github.com/danssampaio/compilers/tree/main/p-code_machine

Realiza a compilação e cria um arquivo executável.
$ gcc p-code.c -o p-code

Inicia o arquivo executável através do terminal.
$ ./p-code

Questões Utilizadas

Para realizar testes nesse programa, foram utilizados as seguintes questões.

1. Soma de dois números inteiros.
2. Soma dos números naturais de 1 até 10.
3. Soma dos números naturais de 1 até 100 (iterativamente).
4. Soma dos quadrados dos números naturais de 1 até 100 (iterativamente).
5. Soma dos cubos dos números naturais de 1 até 100 (iterativamente).

Resultados Obtidos

Questão 1:

INT 0 4
LIT 0 5
LIT 0 3
OPR 0 2
STO 0 3
OPR 0 0

N	Instruction	Level	Argument	StackPtr	ProgCounter	Stack
0	INT	0	4	0	1	[0][0][0][0]
1	LIT	0	5	5	2	[0][0][0][0][5]
2	LIT	0	3	3	3	[0][0][0][0][5][3]
3	OPR	0	2	8	4	[0][0][0][0][8]
4	STO	0	3	8	5	[0][0][0][8]

OUTPUT: 8

INT 0 3
LIT 0 5
LIT 0 3
OPR 0 2
STO 0 2
OPR 0 0

N	Instruction	Level	Argument	StackPtr	ProgCounter	Stack
0	INT	0	3	0	1	[0][0][0]
1	LIT	0	5	5	2	[0][0][0][5]
2	LIT	0	3	3	3	[0][0][0][5][3]
3	OPR	0	2	8	4	[0][0][0][8]
4	STO	0	2	8	5	[0][0][8]

OUTPUT: 8

Questão 2:

```
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
OPR 0 0
```

N	Instruction	Level	Argument	StackPtr	ProgCounter	Stack
0	INT	0	3	0	1	[0][0][0]
1	LIT	0	1	1	2	[0][0][0][1]
2	STO	0	0	0	3	[1][0][0]
3	LIT	0	2	2	4	[1][0][0][2]
4	STO	0	1	0	5	[1][2][0]
5	LOD	0	0	1	6	[1][2][0][1]
.						
.						
.						
96	STO	0	1	45	13	[1][10][45]
97	LOD	0	1	10	14	[1][10][45][10]
98	LOD	0	2	45	15	[1][10][45][10][45]
99	OPR	0	2	55	16	[1][10][45][55]
100	STO	0	2	55	17	[1][10][55]
101	LOD	0	1	10	18	[1][10][55][10]
102	LIT	0	10	10	19	[1][10][55][10][10]
103	OPR	0	7	1	20	[1][10][55][1]
104	JPC	0	9	55	21	[1][10][55]

OUTPUT: 55

Questão 3:

```
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 0 100
OPR 0 7
JPC 0 9
OPR 0 0
```

N	Instruction	Level	Argument	StackPtr	ProgCounter	Stack
0	INT	0	3	0	1	[0][0][0]
1	LIT	0	1	1	2	[0][0][0][1]
2	STO	0	0	0	3	[1][0][0]
3	LIT	0	2	2	4	[1][0][0][2]
4	STO	0	1	0	5	[1][2][0]
5	LOD	0	0	1	6	[1][2][0][1]
6	LOD	0	1	2	7	[1][2][0][1][2]
7	OPR	0	2	3	8	[1][2][0][3]
8	STO	0	2	3	9	[1][2][3]
9	LOD	0	0	1	10	[1][2][3][1]
10	LOD	0	1	2	11	[1][2][3][1][2]
11	OPR	0	2	3	12	[1][2][3][3]
12	STO	0	1	3	13	[1][3][3]
.						
.						
.						
1184	JPC	0	9	5050	21	[1][100][5050]

OUTPUT: 5050

Questão 4:

N	Instruction	Level	Argument	StackPtr	ProgCounter	Stack
0	INT	0	3	0	1	[0][0][0]
1	LIT	0	1	1	2	[0][0][0][1]
2	STO	0	0	0	3	[1][0][0]
3	LIT	0	2	2	4	[1][0][0][2]
4	STO	0	1	0	5	[1][2][0]
5	LOD	0	1	2	6	[1][2][0][2]
6	LOD	0	1	2	7	[1][2][0][2][2]
7	OPR	0	4	4	8	[1][2][0][4]
8	LOD	0	0	1	9	[1][2][0][4][1]
9	OPR	0	2	5	10	[1][2][0][5]

.
.
.

1382	JPC	0	11	338350	25	[1][100][338350]

OUTPUT: 338350

Questão 5:

N	Instruction	Level	Argument	StackPtr	ProgCounter	Stack
0	INT	0	3	0	1	[0][0][0]
1	LIT	0	1	1	2	[0][0][0][1]
2	STO	0	0	0	3	[1][0][0]
3	LIT	0	2	2	4	[1][0][0][2]
4	STO	0	1	0	5	[1][2][0]
5	LOD	0	1	2	6	[1][2][0][2]
6	LOD	0	1	2	7	[1][2][0][2][2]

.
.

.

1580 JPC 0 13 25502500 29 [1][100][25502500]

OUTPUT: 25502500

Considerações Finais

Durante a realização da transcrição do código unicamente foi encontrado uma problemática em relação às funções utilizadas pela linguagem Pascal. Já no código em C, após testar pela primeira vez, o looping infinito surgiu, mas logo foi descoberto qual o problema e resolvido.

Referências

https://en.wikipedia.org/wiki/P-code_machine

https://homepages.cwi.nl/~steven/pascal/book/10pcode.html