Universidade Estadual de Santa Cruz – UESC

Relatório p-code machine utilizando fibonacci e factorial como exemplos

Relatório feito por Darley Souza Sampaio

Disciplina Compiladores

Curso Ciência da Computação

Semestre 2022.2

Professor César 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 utilizadas duas sequências matemáticas solicitadas em sala de aula, o fibonacci de 5 e o factorial de 4.

Fibonacci 5

```
INT 0 3
LIT 0 1
STO 0 0
LIT 0 1
STO 0 1
LIT 0 3
STO 0 2
LOD 0 0
LOD 0 1
OPR 0 2
LOD 0 1
STO 0 0
STO 0 1
LIT 0 5
LOD 0 2
OPR 0 11
JPC 0 22
LOD 0 2
LIT 0 1
OPR 0 2
STO 0 2
JMP 0 7
LOD 0 1
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	1	1	4	[1][0][0][1]
.						
.						
.						
46	JPC	0	22	5	22	[3][5][5]
47	LOD	0	1	5	23	[3][5][5][5]

OUTPUT: 5

Factorial 4

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

```
STO 0 1
LOD 0 1
LOD 0 0
OPR 0 7
JPC 0 7
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	4	4	4	[1][0][0][4]
.						
.						
.						
28	LOD	0	0	1	17	[1][1][24][1][1]
29	OPR	0	7	1	18	[1][1][24][1]
30	JPC	0	7	24	19	[1][1][24]

OUTPUT: 24

Considerações Finais

Durante a realização dos testes dos exemplos utilizados neste programa, não foi encontrada nenhuma dificuldade.

Referências

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

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