

Elementos de Sistemas

Máquina Virtual

"Programadores são criadores de universos da qual só eles são responsáveis. Universos de complexidade praticamente ilimitada podem ser criados sob a forma de programas de computador".

"Programmers are creators of universes for which they alone are responsible. Universes of virtually unlimited complexity can be created in the form of computer programs."

Joseph Weizenbaum (1923-2008), Cientista da Computação

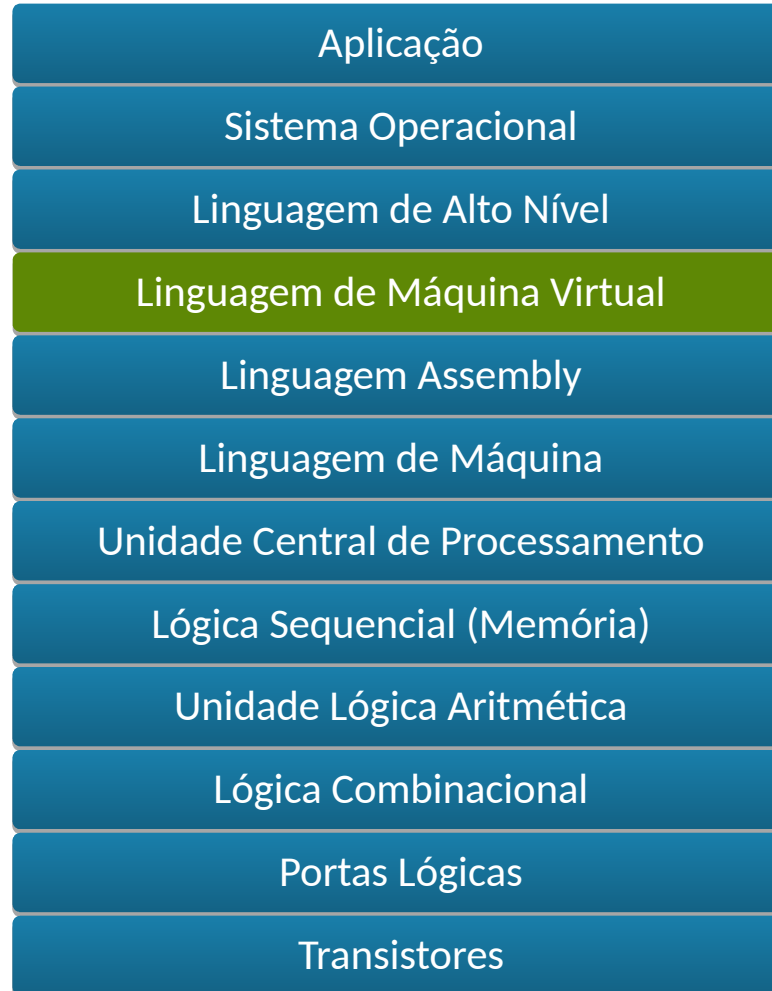
apud Nisan, N. & Schocken, S. 2005. **Elements of Computing Systems**

Objetivos de Aprendizado da Aula

- Distinguir Máquinas Virtuais;
- Trabalhar com Máquina a Pilha.

Conteúdo(s): Interrupções; Autômatos de Pilha;

Níveis de Abstração



Java

JRE

javac

JVM



C#

.NET

csc

CLR

ALGOL e Pascal

ALGOL (Algorithmic Language) iniciou em 1958 em Zurique, sendo uma linguagem que influenciou muitas outras como C e Pascal.

Pascal foi criada na década de 1960 e devido a sua simplicidade e eficiência foi muito usada nas décadas seguintes.

Formar Duplas

James Gosling está querendo começar uma nova plataforma de programação e está procurando pessoa para formar o novo time. Você consegue resolver os problemas de pilha rapidamente?

Formem duplas para resolver as atividades.

Mantenha um ambiente em que todos participem de tudo.



.NET

O framework .NET permite o desenvolvimento em um conjunto de linguagens para rodar em uma máquina virtual.

```
private void btnOpen_Click(object sender, EventArgs e)
{
    string file_name = "C:\\test1.txt";
    string textLine = "";

    System.IO.StreamReader objReader;
    objReader = new System.IO.StreamReader(file_name);

    do
    {
        textLine = textLine + objReader.ReadLine() + "\r\n";
    } while (objReader.Peek() != -1);

    textBox1.Text = textLine;

    objReader.Close();
}
```

CIL

```
0.method private hidebysig
static void 'add'() cil
managed
{
    .maxstack 2
    .locals init ([0] int32 value1,
        [1] int32 value2,
        [2] int32 value3)
    IL_0000: nop
    IL_0001: ldc.i4.s 10
    IL_0003: stloc.0
    IL_0004: ldc.i4.s 20
    IL_0006: stloc.1 0
    IL_0007: ldloc.0
    IL_0008: ldloc.1
    IL_0009: add
    IL_000a: stloc.2
    IL_000b: ret
}
```

Geração de Código

Alto Nível

- Front end
- Tradutor (Compilador + StdLib)

Código Intermediário

- Back end
- Tradução/Interpretação de Código Intermediário

Linguagem de Máquina

Virtual Machine

Dois tipos de máquinas virtuais:

- System Virtual Machine
- Process Virtual Machine
(ou Language Virtual Machine)
(ou Application Virtual Machine)

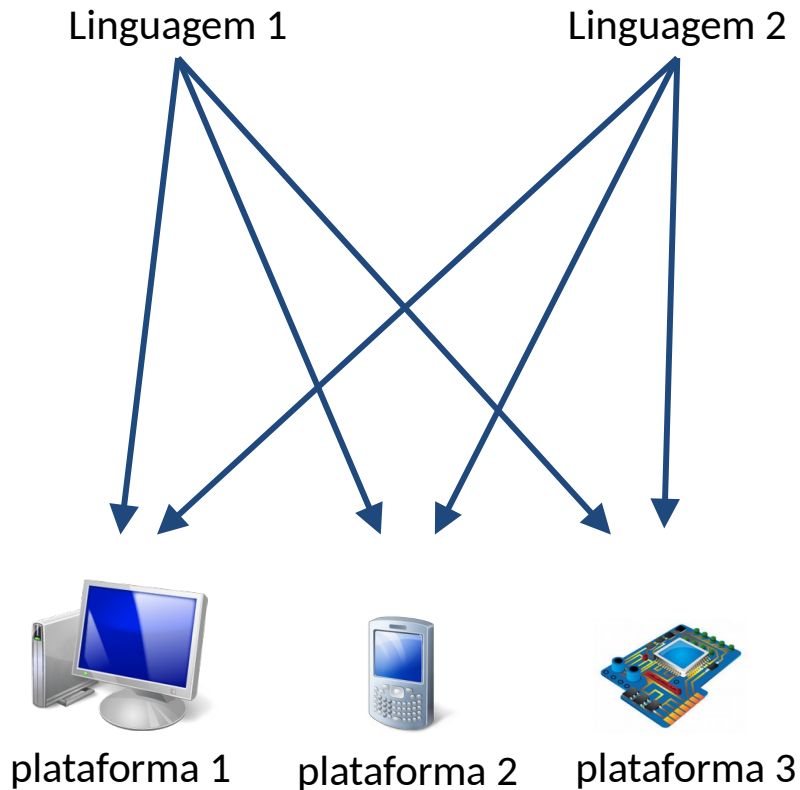
Compiladores

Traduzem programas de alto-nível.

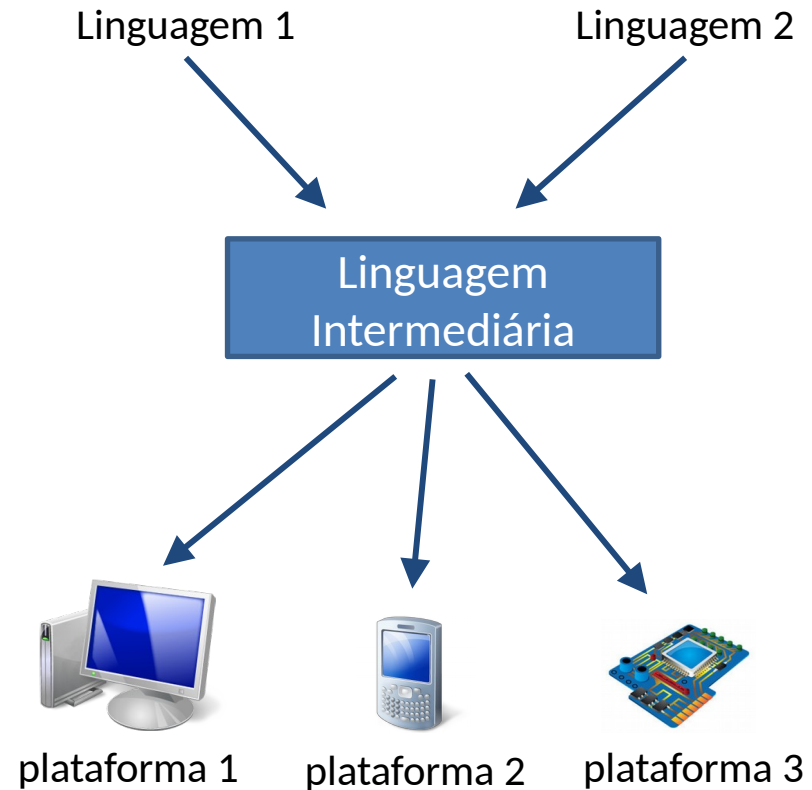
- Diretamente: tem uma forte dependência entre linguagem e arquitetura.
- Por uma linguagem intermediária: Facilita o código ser transportado para outras arquiteturas.

Modelos de Compilação

Compilação Direta (direct compilation)



Compilação de 2 níveis (2-tier compilation)



Compartilhando Backend

Códigos em diferentes linguagem podem compartilhar um backend. Por exemplo:

JVM:

Java, Scala e Clojure

.NET Framework:

C#, F#, Visual Basic.NET

Operações da Linguagem

- Aritméticas
- Acesso a memória
- Fluxo de execução
- Chamada de sub-rotinas

RPN (Reverse Polish Notation)

A Notação Polonesa Inversa, ou notação pós-fixada é usada para calcular operações aritméticas com os operadores após os operandos.

Operação	Notação convencional	Notação Polonesa	Notação Polonesa Inversa
$a + b$	$a+b$	$+ a b$	$a b +$
$\frac{a + b}{c}$	$(a+b)/c$	$/ + a b c$	$a b + c /$
$\frac{a \cdot b - c \cdot d}{e \cdot f}$	$((a*b)-(c*d))/(e*f)$	$/ - * a b * c d * e f$	$a b * c d * - e f * /$

Calculadoras HP

A HP produziu vários calculadores/computadores com a notação polonesa reversa.

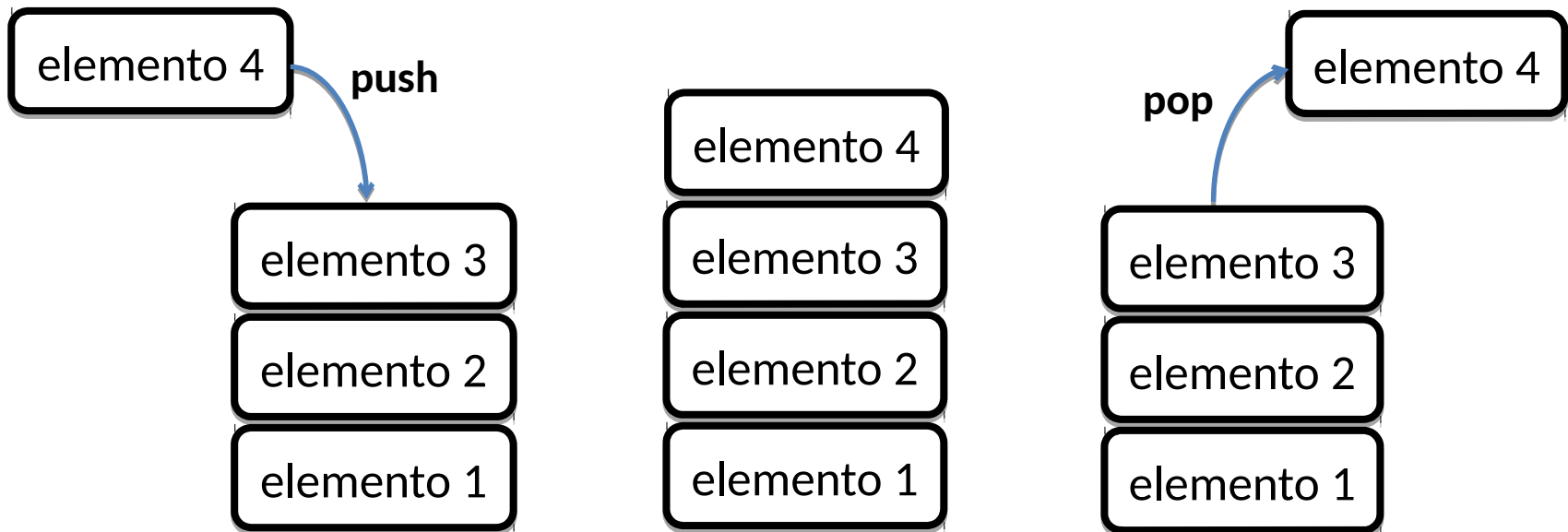


<https://epxx.co/ctb/hp12c.html>

Operações de Pilha

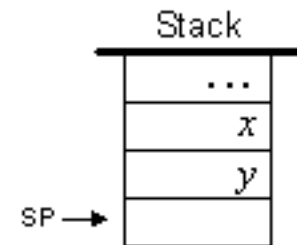
Para o funcionamento de pilhas existem duas operações principais:

- PUSH - insere um elemento no topo da pilha
- POP - retira elemento do topo da pilha.



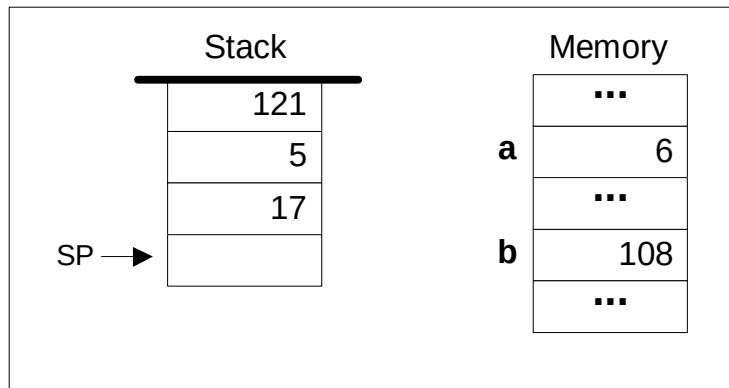
Comandos

Principais comandos em linguagem de pilha:

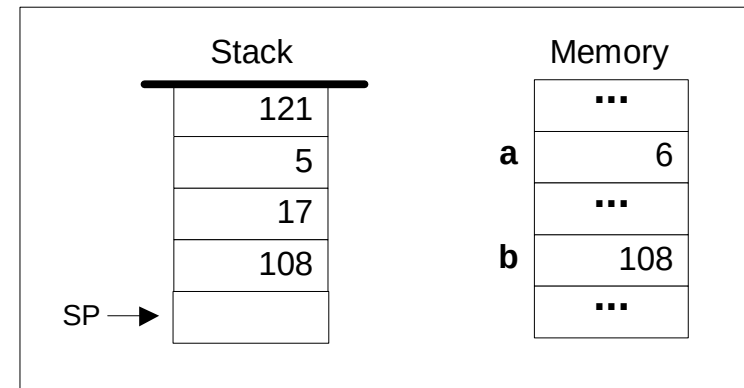
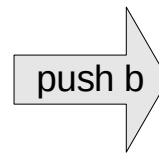


Comando	Operação	Comentário
add	$x + y$	complemento de 2
sub	$x - y$	complemento de 2
neg	$- y$	complemento de 2
eq	$x == y$	igualdade
gt	$x > y$	maior que
lt	$x < y$	menor que
and	$x \& y$	bit-wise
or	$x y$	bit-wise
not	$\sim x$	bit-wise

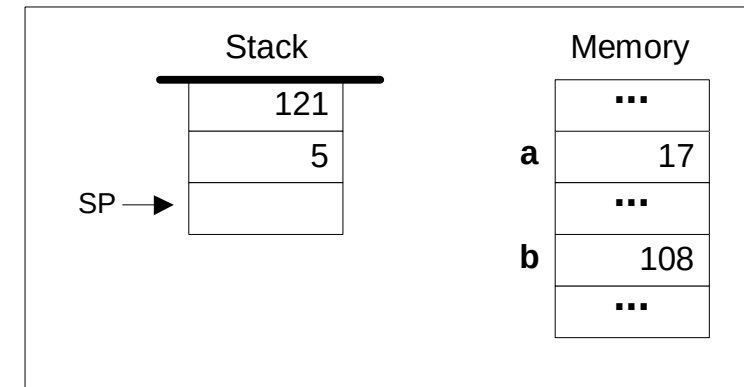
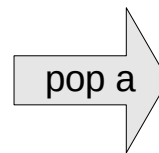
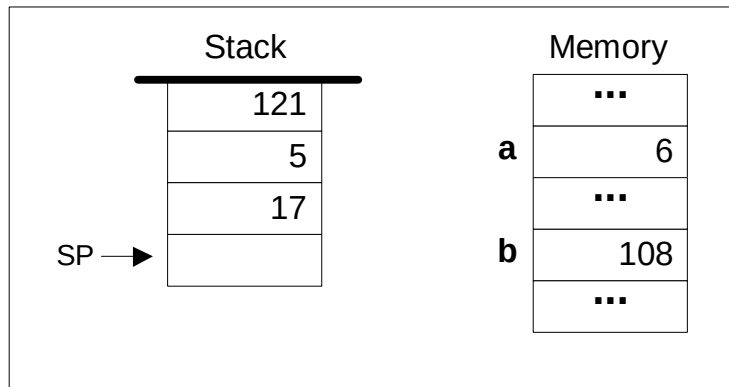
Exemplo de Acesso a Memória



(before)



(after)



Exercício Exemplo

Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

pilha



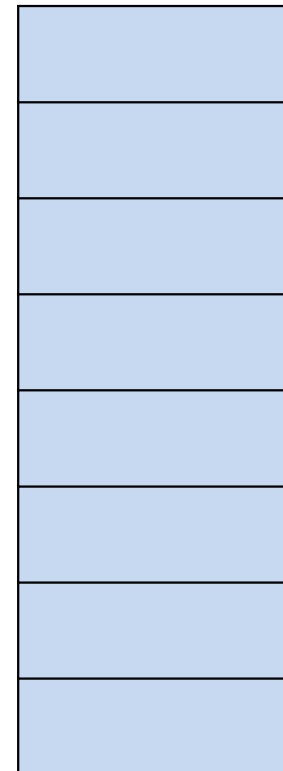
Exercício Exemplo

Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

pilha



Exercício Exemplo

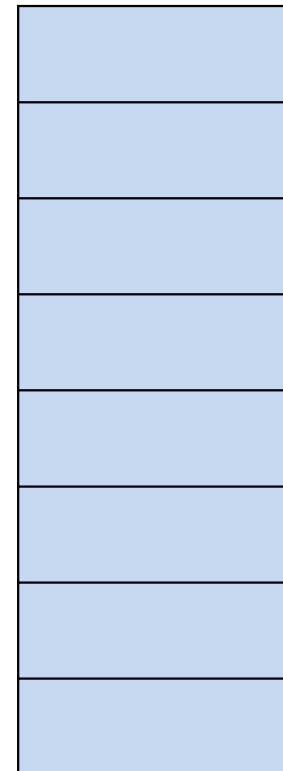
Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

```
push constant 3
neg
push constant 5
push constant 6
add
add
push constant 4
push constant 3
add
sub
```

pilha



Exercício Exemplo

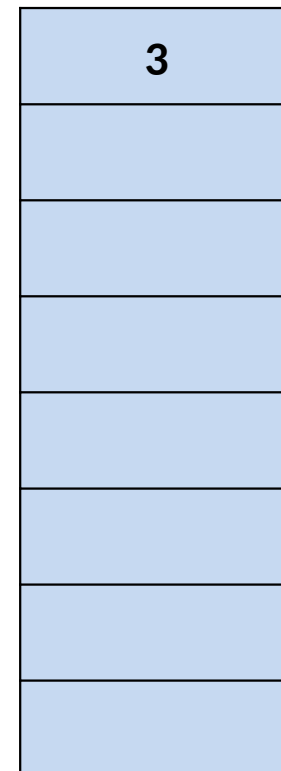
Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

```
push constant 3
neg
push constant 5
push constant 6
add
add
push constant 4
push constant 3
add
sub
```

pilha



Exercício Exemplo

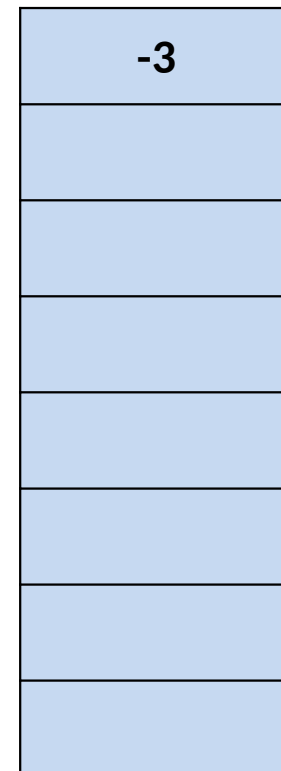
Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

```
push constant 3
neg
push constant 5
push constant 6
add
add
push constant 4
push constant 3
add
sub
```

pilha



Exercício Exemplo

Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

```
push constant 3
neg
push constant 5
push constant 6
add
add
push constant 4
push constant 3
add
sub
```

pilha

-3
5

Exercício Exemplo

Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

```
push constant 3
neg
push constant 5
push constant 6
add
add
push constant 4
push constant 3
add
sub
```

pilha

-3
5
6

Exercício Exemplo

Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

```
push constant 3
neg
push constant 5
push constant 6
add
add
push constant 4
push constant 3
add
sub
```

pilha

-3
11

Exercício Exemplo

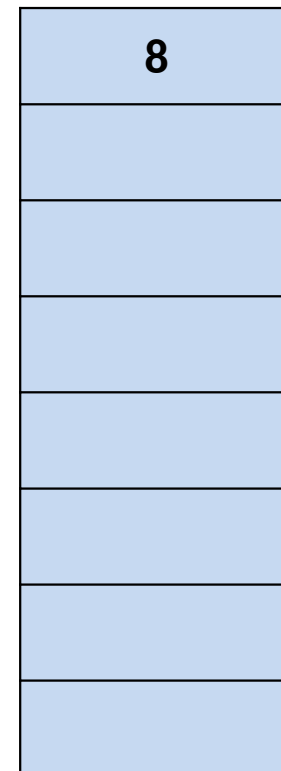
Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

```
push constant 3
neg
push constant 5
push constant 6
add
add
push constant 4
push constant 3
add
sub
```

pilha



Exercício Exemplo

Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

```
push constant 3
neg
push constant 5
push constant 6
add
add
push constant 4
push constant 3
add
sub
```

pilha

8
4

Exercício Exemplo

Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

```
push constant 3
neg
push constant 5
push constant 6
add
add
push constant 4
push constant 3
add
sub
```

pilha

8
4
3

Exercício Exemplo

Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

```
push constant 3
neg
push constant 5
push constant 6
add
add
push constant 4
push constant 3
add
sub
```

pilha

8
7

Exercício Exemplo

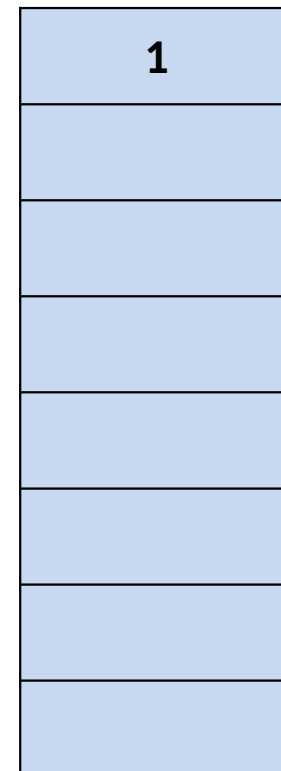
Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem VM do Z0:

$$-3 + (5 + 6) - (4 + 3)$$

RPN = -3 5 6 + + 4 3 + -

```
push constant 3
neg
push constant 5
push constant 6
add
add
push constant 4
push constant 3
add
sub
```

pilha



Exercícios

1) Converta a seguinte equação para a Notação Polonesa Inversa e implemente na máquina de pilha na linguagem de Máquina Virtual:

- a) $(2 + 3 + 4) - (-3 - 4)$
- b) $(2 + 2) - (3 + 3) - (4 + 4)$
- c) $1 + 9 - 5 - (2 + 2)$

2) Converta a seguinte expressão booleana para operações de pilha e implemente na máquina de pilha na linguagem de Máquina Virtual:

- a) $(0 == -1) \text{ AND } (-1 < 0)$
- b) $\text{NOT}(3 > 2 \text{ AND } 2 > 1)$
- c) $(1 \text{ AND } 1) \text{ OR } (0 \text{ AND } 0)$

Exercícios

3) Quais são os resultados dos seguintes programas em linguagem de máquina virtual à pilha:

a)

```
push constant 1
not
push constant 2
add
push constant 4
push constant 4
eq
add
```

b)

c)

```
push constant 0
neg
push constant 5
neg
sub
push constant 5
eq
```

```
push constant 45
push constant 67
push constant 34
push constant 82
push constant 12
push constant 91
add
sub
add
sub
add
```


Próxima Aula

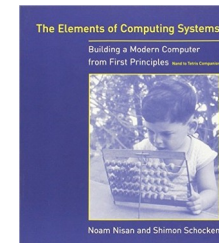
- Ver estudo para aula 18 sobre **Pilhas**
- Estudar Lista de Exercícios Aula 17 (opcional):
- Ler (opcional)

Capítulo 7

The Elements of Computing Systems

Building a Modern Computer from First Principles

Noam Nisan e Shimon Schocken



Insper

www.insper.edu.br

Java

Linguagem de Programação de uso geral orientada a objetos.

Projetada para ser independente de plataforma:

"write once, run anywhere" (WORA)

Java é atualmente uma das linguagens de programação mais populares.



James Gosling