Insper

Elementos de Sistemas

Aula 18 – Operações em Pilhas

"Qualquer tecnologia suficientemente avançada é indistinguível de mágica."
"Any sufficiently advanced technology is indistinguishable from magic."
Arthur C. Clarke (1917-2008), Escritor Britânico

Engenharia - Luciano Soares <lpsoares@insper.edu.br>

Objetivos de Aprendizado da Aula

- Operar em Pilhas;
- Notação Polonesa Reversa;
- Gerenciar Ponteiros de Memória.

Conteúdo(s): Gerenciamento de Memória.

Máquina Virtual do Z0

Para o desenvolvimento do Z0 será adotada uma Maquina Virtual como fase intermediária. Essa máquina virtual pode ser vista como:

- Uma máquina com seu próprio ambiente (um computador virtual).
- Uma representação conveniente de um programa de computador entre o código de alto nível e a linguagem de máquina.

Comandos Suportados pela VM do Z0

Comandos Aritméticos e Booleanos

- **`add`** (soma)
- **`sub`** (subtração)
- **`neg`** (negação)
- **`eq`** (igual)
- **`gt`** (maior que)
- **`lt`** (menor que)
- **`and`** (e)
- **`or`** (ou)
- **`not`** (não)

Comandos de Acesso de Memória

- **`push segmento x`** (empilha)
- **`pop segmento x`** (desempilha)

Comandos de Fluxo de Execução

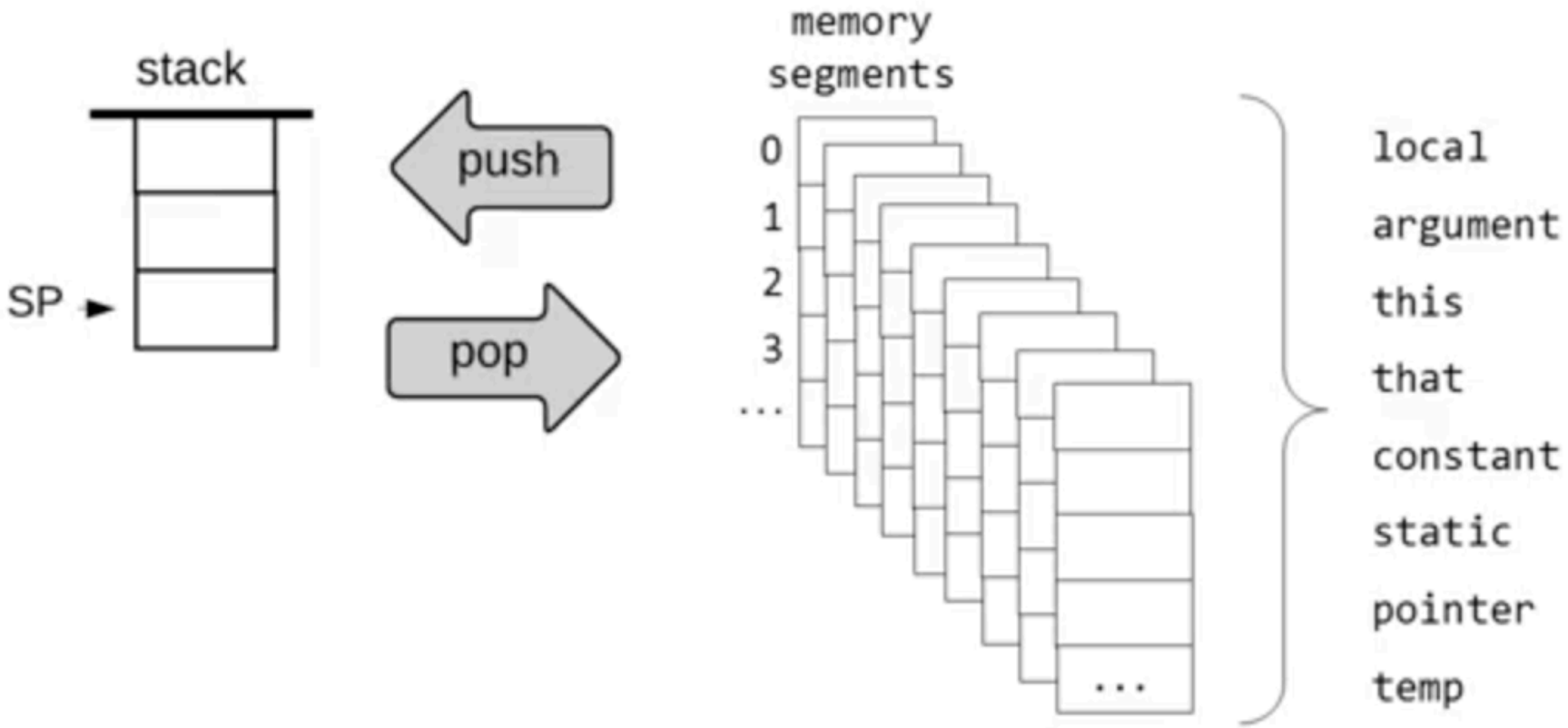
- **`label`** (declaração)
- **`goto`** (marcadores)
- **`if-goto`** (marcadores)

Comandos de Chamada de Funções

- **`function`** (declaração da função)
- **`call`** (invoca uma função)
- **`return`** (retorno da função)

Segmentos de Memória

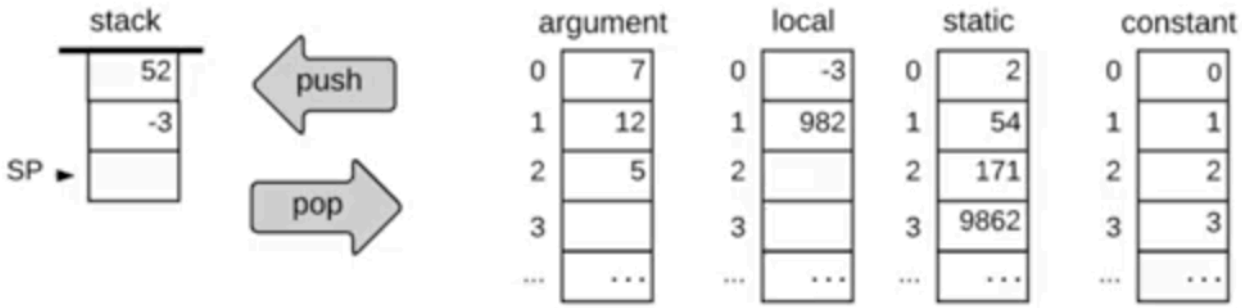
O usuário pode fazer o PUSH ou POP para algum segmento de memória.



Insper

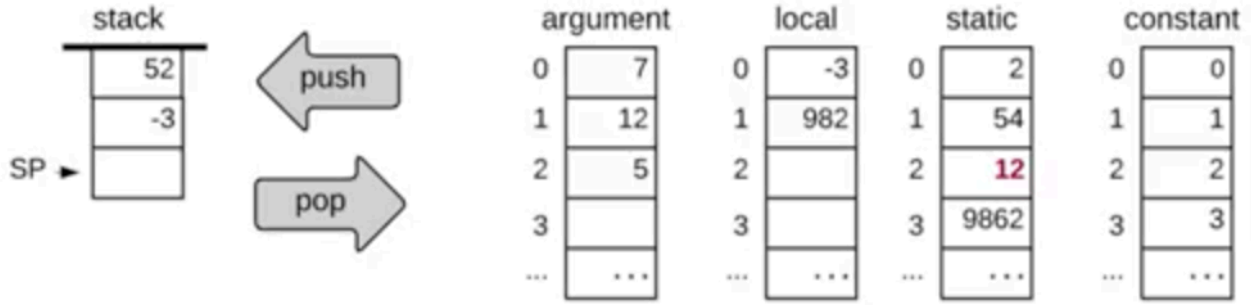
Pergunta

Qual é o código que realiza a operação abaixo?



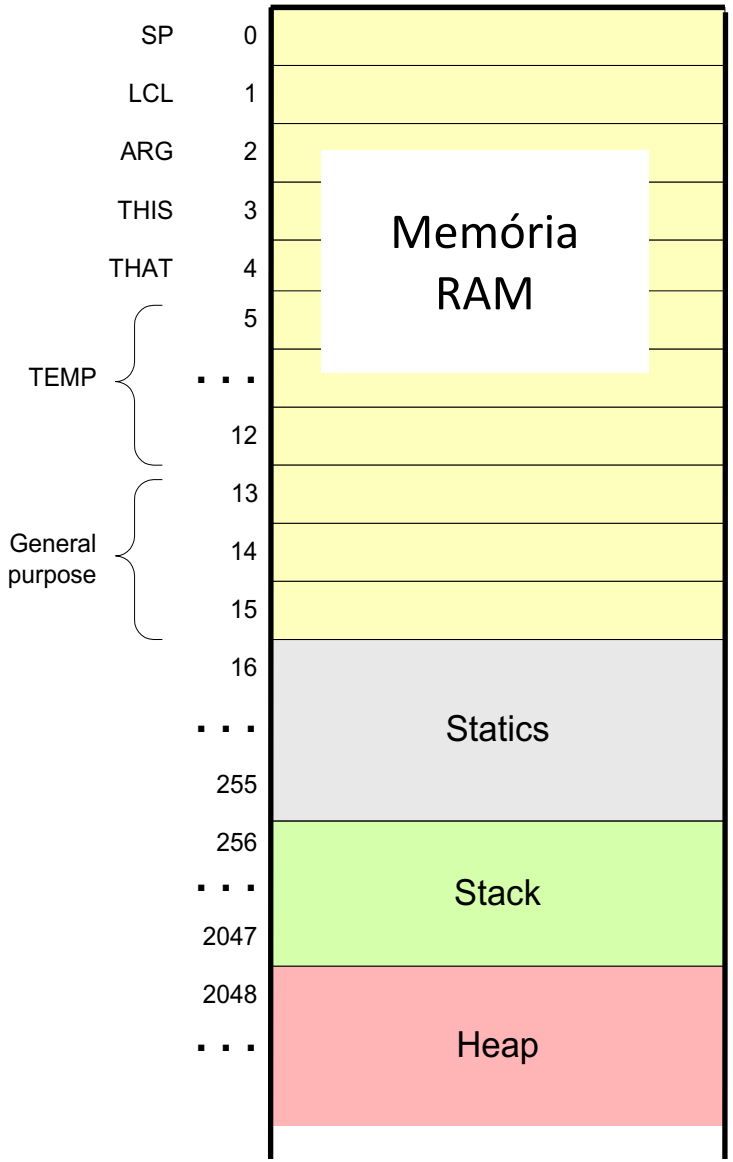
let static 2 = argument 1

?



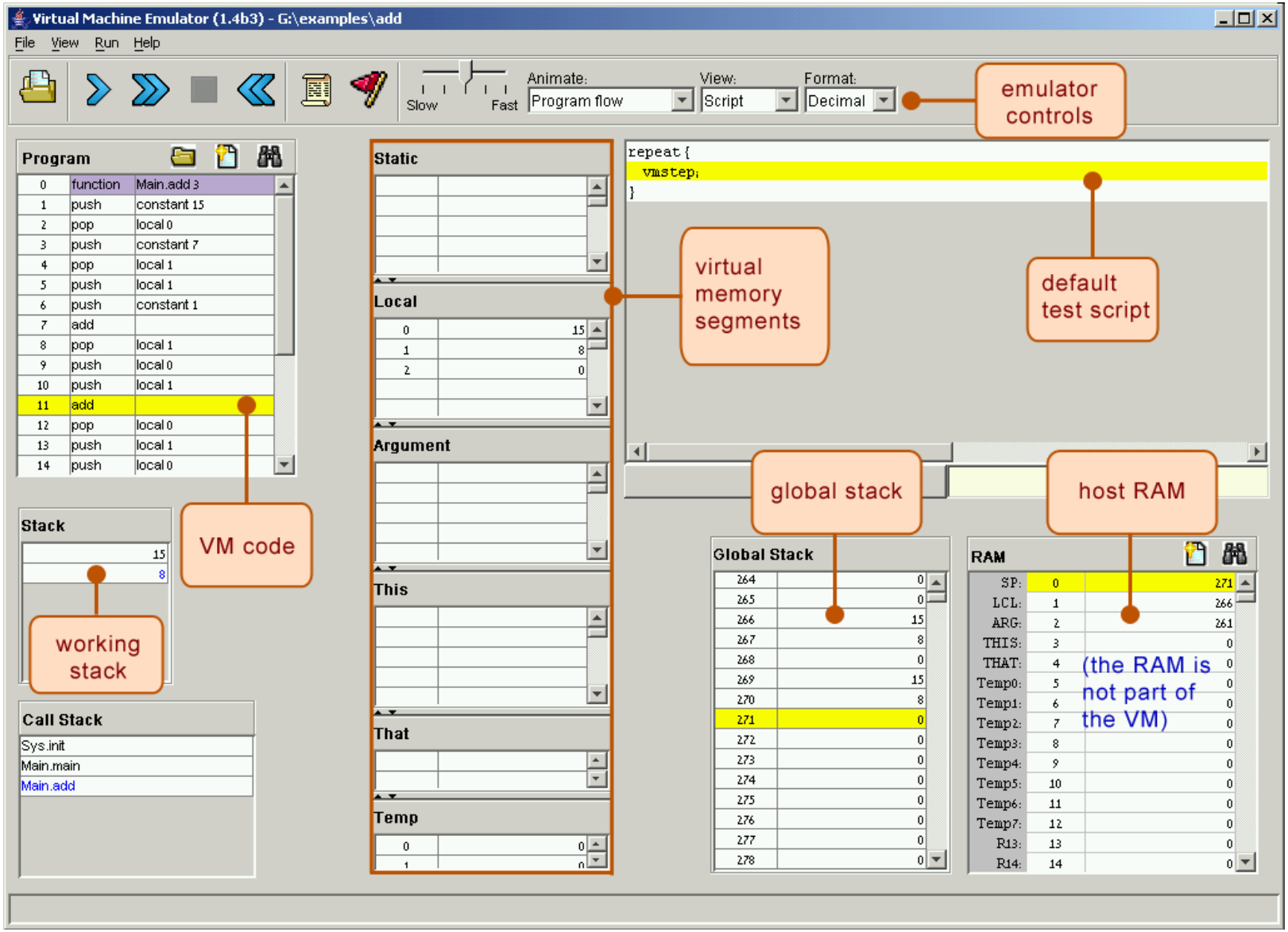
Insper

Organização de memória na VM do Z0



- O **SP** (**S**tack **P**ointer) aponta a posição para a inserção de um novo elemento na pilha (Stack), ele é definido no endereço 0 (Zero) da RAM
- **LCL** (Local), **ARG** (Arguments), **THIS** e **THAT** são ponteiros usados para controlar a posição dos dados nas chamadas de funções.
- A região **Statics** é uma área de memória compartilhada por todas as funções.
- O **Stack** é a pilha de dados do computador. O espaço de memória entre os endereços 256 e 2047 é reservado para armazenar os dados da pilha.
- O **Heap** é a memória dinâmica de dados e pode ser acessada de diversas formas. Essa memória é reservada entre os endereços 2048 e 16383.

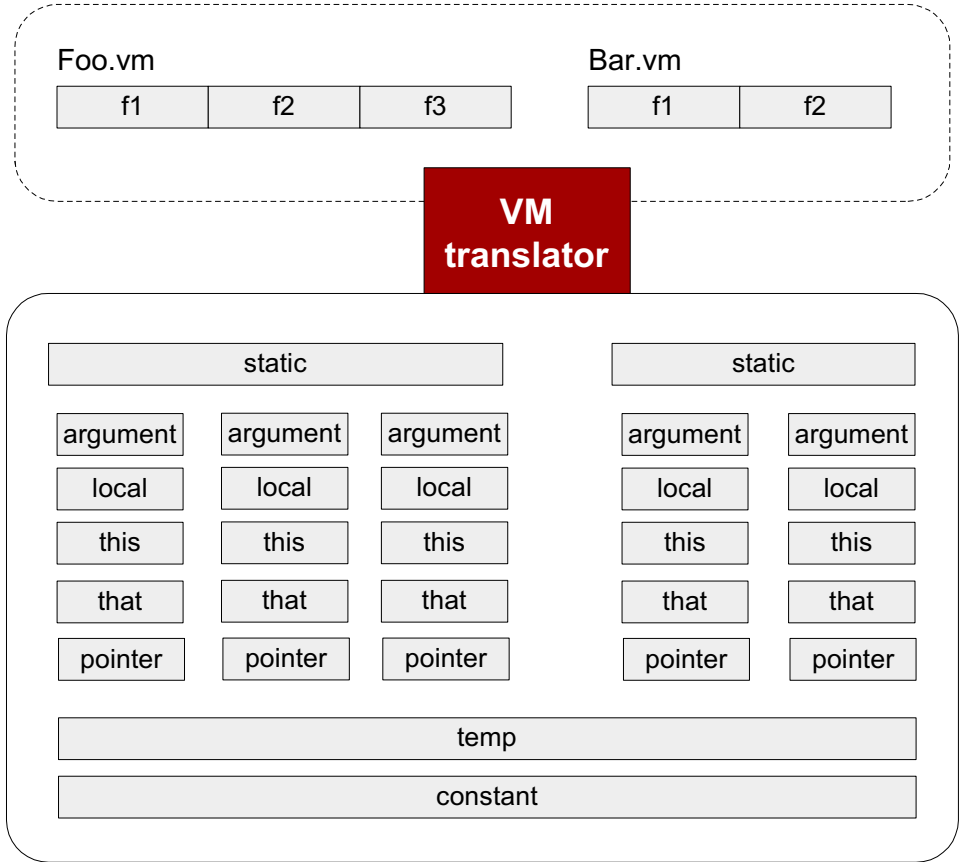
VM Emulator



Insper

Organização dos arquivos

Vários arquivos .vm com suas funções gerando um só .asm



Insper

Manipulação de Ponteiros

Pseudocódigo:

D = *p

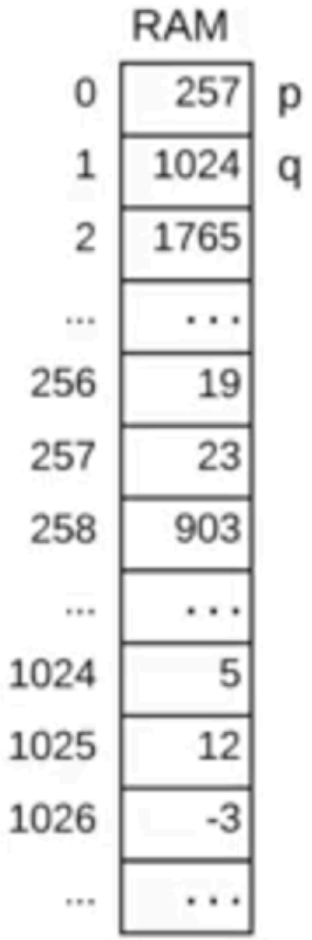
* significa a notação para a localização na memória apontada por p

Em Assembly:

```
leaw $p,%A
movw (%A),%A
movw (%A),%D
```

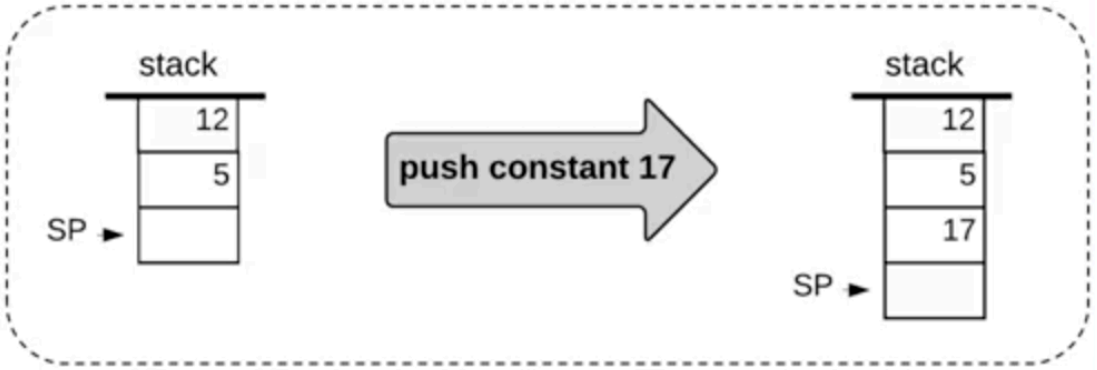
Resultado:

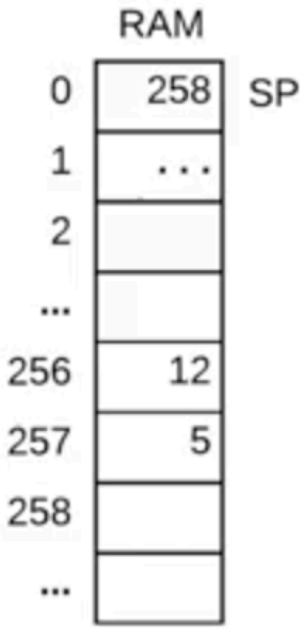
D recebe o valor 23.



Insper

Conversão das instruções



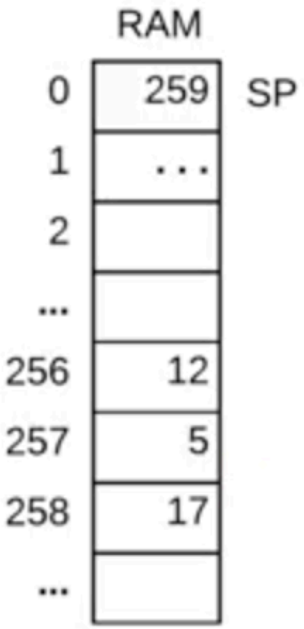


Lógica:

```
*SP = 17
SP++
```

Assembly:

```
leaw $17,%A
movw %A,%D
leaw $SP,%A
movw (%A),%A
movw %D,(%A)
leaw $SP,%A
movw (%A),%D
incw %D
movw %D,(%A)
```



Insper

Funções e Memória

Os programas na máquina virtual são organizados em funções, com cada função organizando seus dados e formas de acesso.

```
function NomeDaFunção nLocals

call NomeDaFunção nArgs

return
```

VEREMOS MAIS TARDE

Dúvidas ?

Lembrem-se:

Tragam sua dúvidas anotadas;

Verifiquem sites como:

- Google
- Stack Overflow
- Etc...

Usem o Slack, para perguntar para seus colegas, ninjas e o professor antes da aula

Formar Duplas

Os Srs. Dennis Ritchie e Ken Thompson estão avaliando criar uma nova linguagem usando um máquina a pilha como linguagem intermediária. Você pode ajudar eles.

Formem duplas para resolver as atividades.

Mantenha um ambiente em que todos participem de tudo.

* Isso é uma obra de ficção, Dennis Ritchie e Ken Thompson não escreveram esse documento.

Exercício

1) Escreve um programa usando as instruções da máquina virtual do Z0 para avaliar qual dos 3 valores nas posições temporárias de memória é o maior, escrevendo ele na posição temp 3 :

```
push constant 3
pop temp 0
push constant 6
pop temp 1
push constant 9
pop temp 2
```

Escreva o programa de forma que seja possível trocar os valores e o resultado continuar sendo correto.

Exercício

2) Implemente as seguintes instruções da máquina virtual em Assembly do Z0:

```
a) push constant 54

b) pop temp 0

c) and

d) add
```

Próxima Aula

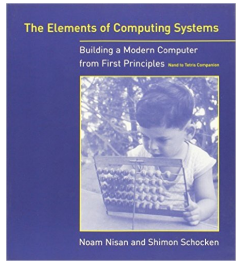
- Ver estudo para aula ? sobre **????**
- Estudar Lista de Exercícios Aula ? (opcional):
- Ler (opcional)

Capítulo 8

The Elements of Computing Systems

Building a Modern Computer from First Principles

Noam Nisan e Shimon Schocken



Insper

Insper

www.insper.edu.br