Insper

Elementos de Sistemas

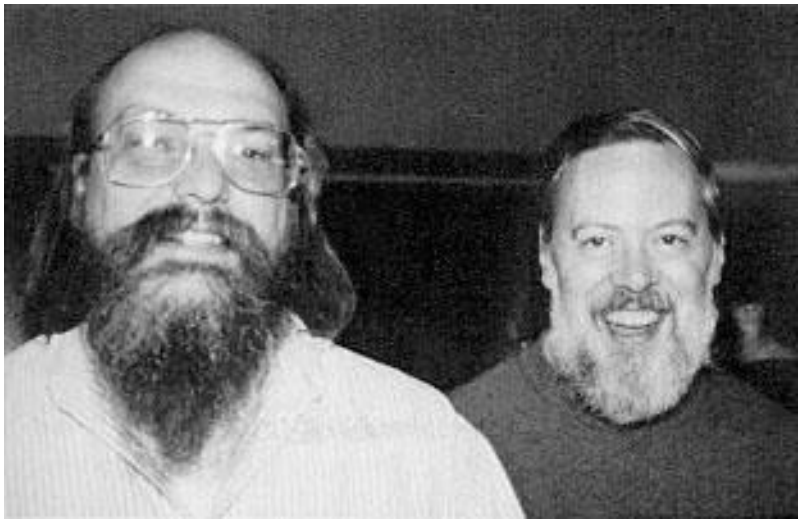
Operações em Pilhas

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

Ciência da Computação - Luciano Soares <lpsoares@insper.edu.br>

Ken Thompson e Dennis Ritchie

Ken Thompson e Dennis Ritchie participaram da criação do Unix nos Bell Labs. Ritchie desenvolveu C a partir da linguagem B de Thompson.



Ken Thompson Dennis Ritchie

Lua

Lua é uma linguagem de script compacta, que suporta programação procedural, programação orientada a objetos, programação funcional, dentre outras.

Programas Lua são executados através da interpretação de um bytecode com uma máquina virtual baseada em registradores. A linguagem é muito usada para arquivos de configuração, scripts e prototipagem rápida.

Lua exemplifica uma VM baseada em registradores; a VM que estudaremos é baseada em pilha.



Waldemar, Roberto, Luiz

Motivação

```
class Main {
  static int x;

  function void main() {
    // Input and multiply 2 numbers
    var int a, b, x;
    let a = Keyboard.readInt("Enter a number");
    let b = Keyboard.readInt("Enter a number");
    let x = mult(a,b);
    return;
  }

  // Multiplies two numbers.
  function int mult(int x, int y) {
    var int result, j;
    let result = 0; let j = y;
    while (~(j = 0)) {
      let result = result + x;
      let j = j - 1;
    }
    return result;
  }
}
```

Traduzir programas em linguagem de alto nível para um código executável.

Compilar

```
    ...
    leaw $a,%A
    movw %D,(%A)
    leaw $b,%A
    movw $0,(%A)
LOOP:
    leaw $a,%A
    movw (%A),%D
    leaw $b,%A
    subw %D,%A,%D
    leaw $END,%A
    jg
    leaw $j,%A
    movw (%A),%D
    leaw $temp,%A
    addw %D,(%A),%D
    movw %D,(%A)
    leaw $j,%A
    movw (%A),%D
    incw %D
    movw %D,(%A)
    leaw $LOOP,%A
    jmp
    ...
```

Aritmética por Pilha

Revendo:



Insper

Stack, SP e Heap

Stack (pilha) é uma região de memória organizada em ordem LIFO, utilizada pela VM para operandos e resultados intermediários.

SP é um ponteiro que mostra o próximo local de memória disponível para armazenar um valor. As operações de push e pop fazem esse ponteiro aumentar e diminuir.

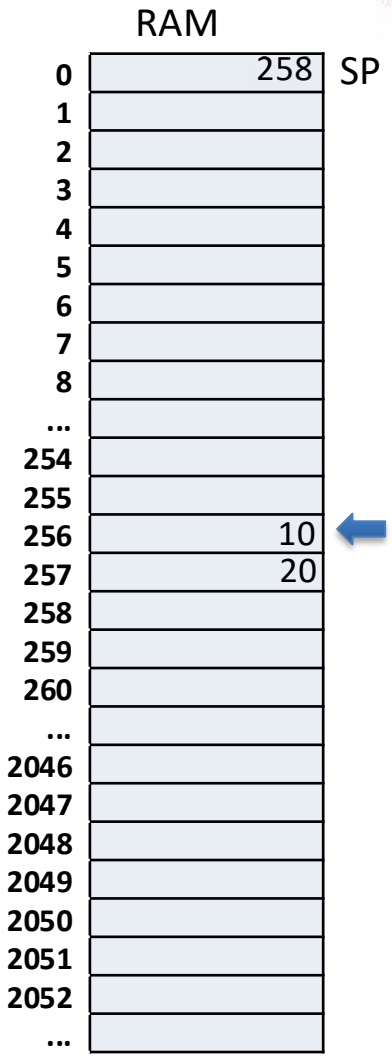
Heap é a área usada para armazenar dados dinâmicos diversos dos programas. Os programas alocam e desalocam memória para seu funcionamento aqui.

Exemplo de Memória

Quando uma constante é inserida por um PUSH, ela é armazenada diretamente na pilha.

Exemplo:

```
push constant 10
push constant 20
```



Insper

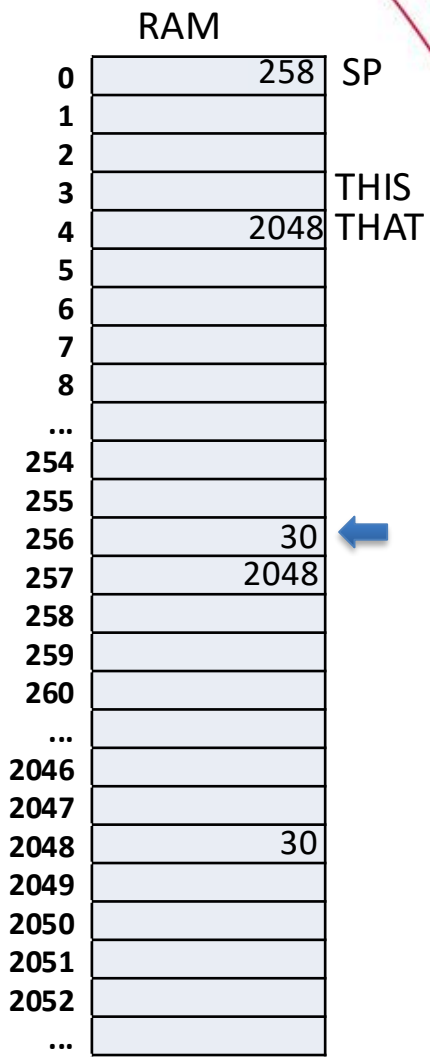
Exemplo de Memória

O *heap* é acessado pelo THIS e THAT.

As posições 0 e 1 do segmento pointer permitem acessar e modificar THIS e THAT, respectivamente.

Exemplo:

```
push constant 30
push constant 2048
pop pointer 1
pop that 0
```



Insper

Vetores (Arrays)

Vetores são espaços de memória usados para alocar uma sequencia de valores. Para acessar uma determinada célula basta somar o endereço base ao índice da célula desejada. Por exemplo:

```
push constant 2048  //endereço base
pop temp 0
push constant 0  // índice
pop temp 1
label loop
push temp 0
push temp 1
add
pop pointer 0
push temp 1
pop this 0
push temp 1
push constant 1
add
pop temp 1
goto loop
```

Obs: exemplo de acesso sem verificação de limites

Insper

Fluxo de Execução

Os programas na máquina virtual navegam por labels e comandos de goto.

label rótulo

goto rótulo

if-goto rótulo

if-goto rótulo retira o valor do topo da pilha. Se esse valor for diferente de zero, desvia para o rótulo; caso contrário, executa a próxima instrução.

```
   push constant 2
   pop temp 0
   push constant 4
   pop temp 1
   push constant 0
   pop temp 2
label loop
   push constant 0
   push temp 1
   eq
   if-goto end
   push temp 0
   push temp 2
   add
   pop temp 2
   push temp 1
   push constant 1
   sub
   pop temp 1
   goto loop
label end
```


Insper

Usando GOTO na Máquina Virtual

```
if (cond)
   s1
else
   s2
...
```

```
  VM code for computing ~(cond)
  if-goto L1
  VM code for executing s1
  goto L2
label L1
  VM code for executing s2
label L2
  ...
```

```
while (cond)
   s1
...
```

```
label L1
  VM code for computing ~(cond)
  if-goto L2
  VM code for executing s1
  goto L1
label L2
  ...
```

Programação em Máquinas Virtuais

Programas em linguagem de Máquinas Virtuais são normalmente escritos por compiladores, não por seres humanos;

Mas, para um ser humano escrever ou otimizar um compilador, ele deve primeiro entender o espírito da programação em máquinas virtuais.

Insper

www.insper.edu.br