

# Máquina Virtual

## Avaliação de Expressões

o valor lógico verdadeiro é avaliado como 1 e falso como 0.

## Instruções

<del>LDC</del>	<del><math>k</math></del>	(Carregar constante): <del><math>S := s + 1 ; M[s] := k</math></del>
<del>LDV</del>	<del><math>n</math></del>	(Carregar valor): <del><math>S := s + 1 ; M[s] := M[n]</math></del>
<del>ADD</del>		(Somar): <del><math>M[s+1] := M[s+1] + M[s] ; s := s + 1</math></del>
<del>SUB</del>		(Subtrair): <del><math>M[s+1] := M[s+1] - M[s] ; s := s + 1</math></del>
<del>MULT</del>		(Multiplicar): <del><math>M[s+1] := M[s+1] * M[s] ; s := s + 1</math></del>
<del>DIV</del>		(Dividir): <del><math>M[s+1] := M[s+1] \text{ div } M[s] ; s := s + 1</math></del>
<del>INV</del>		(Inverter sinal): <del><math>M[s] := -M[s]</math></del>
<del>AND</del>		(Conjunção): <del>se <math>M[s+1] = 1</math> e <math>M[s] = 1</math> então <math>M[s+1] := 1</math> senão <math>M[s+1] := 0 ; s := s + 1</math></del>
<del>OR</del>		(Disjunção): <del>se <math>M[s+1] = 1</math> ou <math>M[s] = 1</math> então <math>M[s+1] := 1</math> senão <math>M[s+1] := 0 ; s := s + 1</math></del>
<del>NEG</del>		(Negação): <del><math>M[s] := 1 - M[s]</math></del>
<del>CME</del>		(Comparar menor): <del>se <math>M[s+1] &lt; M[s]</math> então <math>M[s+1] := 1</math> senão <math>M[s+1] := 0 ; s := s + 1</math></del>
<del>CMA</del>		(Comparar maior): <del>se <math>M[s+1] &gt; M[s]</math> então <math>M[s+1] := 1</math> senão <math>M[s+1] := 0 ; s := s + 1</math></del>
<del>CEQ</del>		(Comparar igual): <del>se <math>M[s+1] = M[s]</math> então <math>M[s+1] := 1</math> senão <math>M[s+1] := 0 ; s := s + 1</math></del>
<del>CDIF</del>		(Comparar desigual): <del>se <math>M[s+1] \neq M[s]</math> então <math>M[s+1] := 1</math> senão <math>M[s+1] := 0 ; s := s + 1</math></del>
<del>CMEQ</del>		(Comparar menor ou igual): <del>se <math>M[s+1] \leq M[s]</math> então <math>M[s+1] := 1</math> senão <math>M[s+1] := 0 ; s := s + 1</math></del>
<del>CMAQ</del>		(Comparar maior ou igual): <del>se <math>M[s+1] \geq M[s]</math> então <math>M[s+1] := 1</math> senão <math>M[s+1] := 0 ; s := s + 1</math></del>
<del>START</del>		(Iniciar programa principal): <del><math>S := 1</math></del>
<del>HLT</del>		(Parar): <del>“Pára a execução da MVD”</del>

### Atribuição

STR  $n$  (Armazenar valor):

$M[n] := M[s]; s := s - 1$

### Desvios (não há o incremento implícito sobre $i$ )

JMP  $t$  (Desviar sempre):

$i := t$

JMPF  $t$  (Desviar se falso):

se  $M[s] = 0$  então  $i := t$  senão  $i := i + 1$ ;

$s := s - 1$

### Operação Nula

NULL (Nada)

### ~~Entrada~~

~~RD (Leitura):~~

~~$S := s + 1; M[s] := \text{"próximo valor de entrada"}.$~~

### ~~Saída~~

~~PRN (Impressão):~~

~~"Imprimir  $M[s]$ ";  $s := s - 1$~~

### Alocação e Desalocação de Variáveis

ALLOC  $m, n$  (Alocar memória):

**Para**  $k := 0$  até  $n - 1$  **faça**

$\{ s := s + 1; M[s] := M[m + k] \}$

DALLOC  $m, n$  (Desalocar memória):

**Para**  $k := n - 1$  até  $0$  **faça**

$\{ M[m + k] := M[s]; s := s - 1 \}$

### Chamada de Rotina

CALL  $t$  (Chamar procedimento ou função):

$S := s + 1; M[s] := i + 1; i := t$

RETURN (Retornar de procedimento):

$i := M[s]; s := s - 1$

**Exemplo:**

```
programa exemplo6;  
  var x, y: integer;  
  
  procedimento p;  
    var z: integer;  
  inicio  
    z:= x; x:=x-1;  
    se z>1 entao p (1)  
    senao y:=1;  
    y:=y*z  
  fim { p };  
inicio  
  leia(x);  
  p;  
  escreva (y);  
  escreva (x)  
fim.
```

**A pilha *M* está invertida!**

**Supondo que o valor lido em x seja igual a 2:**

	START		<b>programa</b>									
	ALLOC	0,2	<b>var x,y</b>		x		x		x		x	
	JMP	L1										
L2	NULL		<b>procedimento p</b>									
	ALLOC	2,1	<b>var z</b>		y		y		y		y	
	LDV	0										
	STR	2	<b>z:=x</b>									
	LDV	0				z1	z	z2	z	z3	z	
	LDC	1										
	SUB							*		*	*	
	STR	0	<b>x:=x-1</b>					*		*	*	
	LDV	2						*		*	*	
	LDC	1										
	CMA		<b>se z&gt;1</b>					z1		z1	z1	
	JMPF	L3	<b>entao</b>									
	CALL	L2	<b>p</b>							*	*	
	JMP	L4								*	*	
L3	NULL		<b>senao</b>							*	*	
	LDC	1										
	STR	1	<b>y:=1</b>							z2	z2	
L4	NULL											
	LDV	1									*	
	LDV	2									*	
	MULT										*	
	STR	1	<b>y:=y*z</b>									
	DALLOC	2,1	<b>fim</b>									
	RETURN										z3	
L1	NULL											
	RD											
	STR	0	<b>leia(x)</b>									
	CALL	L2	<b>p</b>									
	LDV	0										
	PRN		<b>escreva (x)</b>									
	LDV	1										
	PRN		<b>escreva (y)</b>									
	DALLOC	0,2	<b>fim.</b>									
	HLT											

## Exemplo de interface para a Máquina Virtual:

