

# Máquina Virtual

## Características Gerais da MVD

A *região de programa  $P$*  conterá as instruções para a MVD. O formato exato de cada instrução é irrelevante, neste momento, para a nossa discussão.

A *região da pilha de dados  $M$*  conterá os valores manipulados pelas instruções da MVD. Suporemos que esta região compõe-se de palavras que podem conter valores inteiros ou então indefinidos.

Suporemos que cada uma das três regiões têm palavras endereçadas (indexadas) por 0, 1, 2,..., e não nos preocuparemos com as limitações de tamanho de cada região, nem de cada palavra.

A MVD terá dois registradores especiais que serão usados para descrever o efeito das instruções:

O registrador do programa  $i$  conterá o endereço da próxima instrução a ser executada, que será, portanto,  $P [i]$ .

O registrador  $s$  indicará o elemento no topo da pilha cujo valor será dado, portanto, por  $M [s]$ .

A execução de cada instrução incrementa de um o valor de  $i$ , exceto as instruções que envolvem desvios  $i:=i+1$  (*implícito*).

## Avaliação de Expressões

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

## Instruções

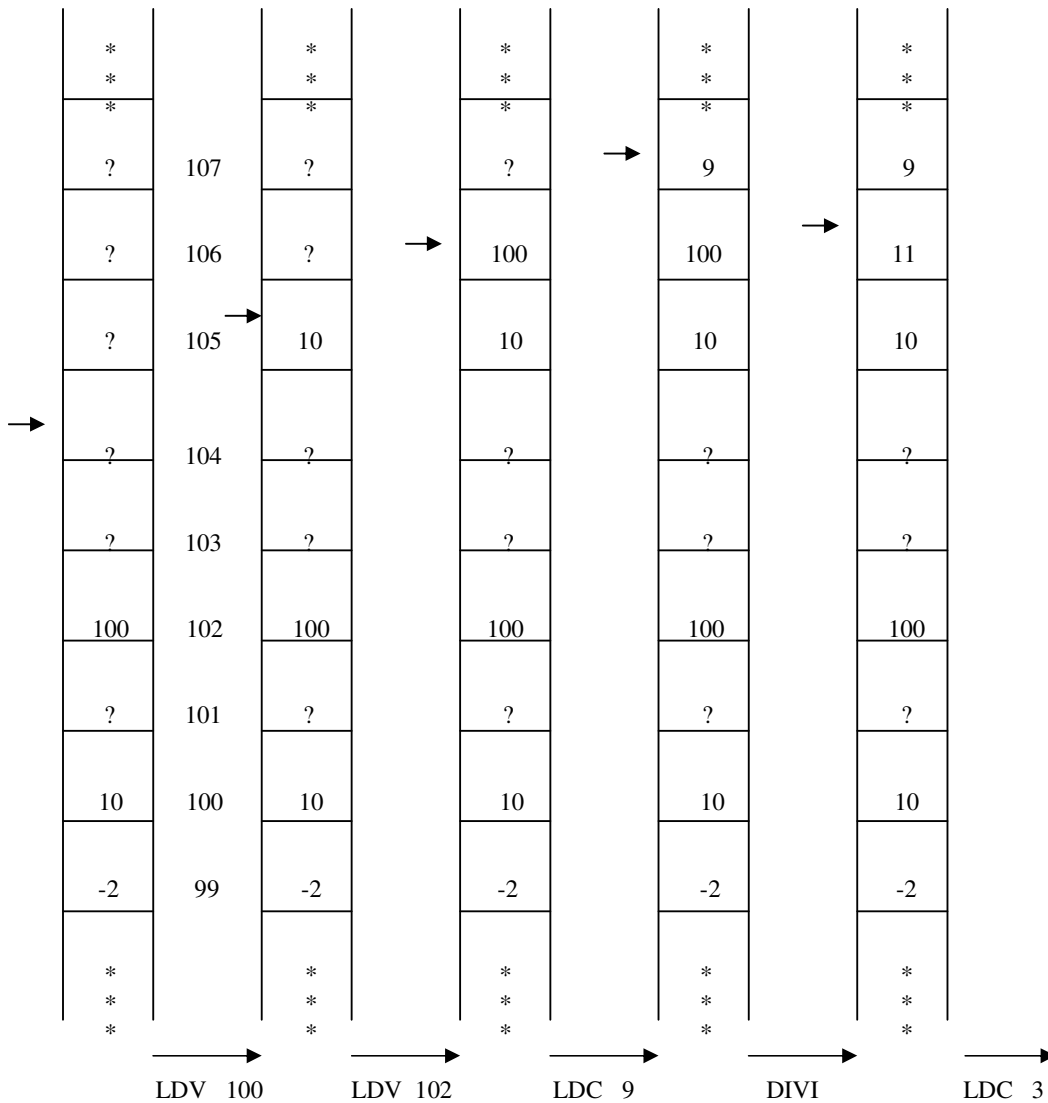
LDC	$k$	(Carregar constante): $S := s + 1 ; M[s] := k$
LDV	$n$	(Carregar valor): $S := s + 1 ; M[s] := M[n]$
ADD		(Somar): $M[s-1] := M[s-1] + M[s] ; s := s - 1$
SUB		(Subtrair): $M[s-1] := M[s-1] - M[s] ; s := s - 1$
MULT		(Multiplicar): $M[s-1] := M[s-1] * M[s] ; s := s - 1$
DIVI		(Dividir): $M[s-1]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

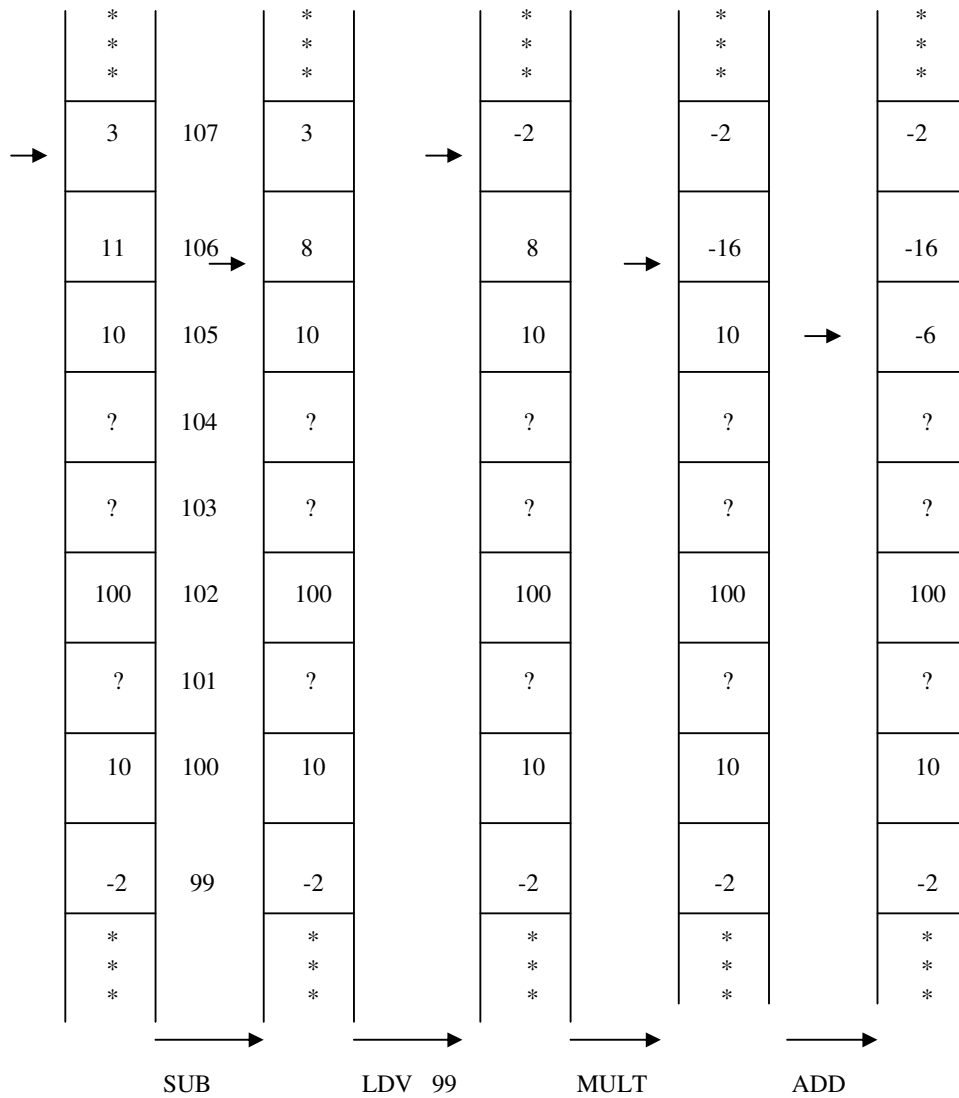
**Exemplo:**  $a + (b \text{ div } 9 - 3) * c$ ,  
 a, b e c estão, respectivamente, nas posições 100, 102, e 99.

ab9 div 3-c\*+.

```
LDV 100
LDV 102
LDC 9
DIVI
LDC 3
SUB
LDV 99
MULT
ADD
```

Suponhamos que os valores armazenados nas posições 99, 100 e 102 da pilha sejam -2, 10 e 100, respectivamente, e que o registrador s contém o valor 104.





**Atribuição**

STR *n* (Armazenar valor):

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

**Exemplo:**

a := a + b \* c

```
LDV 100
LDV 102
LDV 99
MULT
ADD
STR 100
```

### 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)

#### se *E* então *C* 1 senao *C* 2

```

    *
    *           tradução de E
    *
JMPF / 1
    *
    *           tradução de C 1
    *
JMP / 2
11 NULL
    *
    *           tradução de C 2
    *
/2 NULL
```

#### se *E* então *C*

```

    *
    *           tradução de E
    *
JMPF /
    *
    *           tradução de C
    *
/ NULL
```

#### enquanto *E* faça *C*

```

/1 NULL
    *
    *           tradução de E
    *
JMPF /2
    *
    *           tradução de C
    *
JMP /1
/2 NULL
```

**Exemplo:**

1. **se q entao** a:=1 **senao** a:=2

```
LDV Q
JMPF L1
LDC 1
STR A
JMP L2
L1 NULL
LDC 2
STR A
L2 NULL
```

2. **enquanto** s<=n **faca** s:=s+3\*s

```
L7 NULL
LDV S
LDV N
CMEQ
JMPF L8
LDV S
LDC 3
LDV S
MULT
ADD
STR S
JMP L7
L8 NULL
```

**Entrada**

RD (Leitura):  
S:=s + 1; M[s]:= “próximo valor de entrada”.

**Exemplo:**

leia(v1):

```
RD
STR V1
```

Onde V1 é o endereço da variável v1.

**Saída**

PRN (Impressão):  
“Imprimir M[s]”; s:=s-1

**Exemplo:**

escreva (v1):

```
LDV V1
PRN
```

### 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	x	x	x	x
JMP	L1									
L2 NULL		<b>procedimento p</b>								
ALLOC	2,1	<b>var z</b>	y	y	y	y	y	y	y	y
LDV	0									
STR	2	<b>z:=x</b>								
LDV	0		z1	z	z2	z	z3	z		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										

**Exercício:** Implementar retorno de função usando o comando RETURN (o mesmo usado para procedimento).



## Exemplo de interface para a Máquina Virtual:

**Simulador**  
Arquivos Executar Sobre...

*Instruções a serem executadas pela MV*

I	Instrução	Atributo #1	Atributo #2	Comentário
56	RETURN			
57	NULL			Não faz nada
58	RD			$S = S + 1$ ; $M[3] = 2$ ;
59	STR	0		$M[0] = M[3]$ ; $S = S - 1$
60	LDV	0		$S = S + 1$ ; $M[3] = M[0]$ ;
61	LDC	1		$S = S + 1$ ; $M[4] = 1$ ;
62	CEQ			Se ( $M[3] = M[4]$ ) então $M[3]=1$ ; senão $M[3]=0$ ; $S = S - 1$
63	JMPF	66		Se $M[3] = 0$ então $I = 66$ ; senão $I = I + 1$ ; $S = S - 1$ ;
64	CALL	4		
65	JMP	72		
66	NULL			Não faz nada

*Conteúdo da Pilha*

Endereço (S)	Valor
0	2
1	
2	

*Janela de Entrada*

2  
4  
2  
4  
21  
1  
?

*Janela de Saída*

24  
24  
1

*Break Point's*

Continuar

Programa sendo executado na maquina virtual didática. MODDO DE EXECUÇÃO : Passo a passo. I = 66

Pág 6 Seção 1 6/6 Em Lin Col GRA AUT EST SE