

# MÁQUINA VIRTUAL

CÓDIGO	parâmetro	comentário
aritméticas		
ADD	0	executar operação aritmética soma
DIV	0	executar operação aritmética divisão
MUL	0	executar operação aritmética multiplicação
SUB	0	executar operação aritmética subtração
memória (variáveis, constantes)		
ALB	deslocamento	alocar espaço na pilha de dados, para variáveis do tipo lógico, igual ao <b>deslocamento</b> passado como parâmetro
ALI	deslocamento	alocar espaço na pilha de dados, para variáveis do tipo inteiro, igual ao <b>deslocamento</b> passado como parâmetro
ALR	deslocamento	alocar espaço na pilha de dados, para variáveis do tipo real, igual ao <b>deslocamento</b> passado como parâmetro
ALS	deslocamento	alocar espaço na pilha de dados, para variáveis do tipo literal, igual ao <b>deslocamento</b> passado como parâmetro
LDB	constante	carregar na pilha de dados a <b>constante</b> lógica passada como parâmetro
LDI	constante	carregar na pilha de dados a <b>constante</b> inteira passada como parâmetro
LDR	constante	carregar na pilha de dados a <b>constante</b> real passada como parâmetro
LDS	constante	carregar na pilha de dados a <b>constante</b> literal passada como parâmetro
LDV	endereço	carregar na pilha de dados o valor armazenado no <b>endereço</b> passado como parâmetro
STR	endereço	armazenar conteúdo do topo da pilha de dados no <b>endereço</b> passado como parâmetro
lógica		
AND	0	executar operação lógica E
NOT	0	executar operação lógica NÃO
OR	0	executar operação lógica OU
relacional		
BGE	0	executar operação relacional maior ou igual
BGR	0	executar operação relacional maior que
DIF	0	executar operação relacional diferente
EQL	0	executar operação relacional igual
SME	0	executar operação relacional menor ou igual
SMR	0	executar operação relacional menor que
desvio / controle		
JMF	endereço	desviar quando for falso para a instrução do <b>endereço</b> passado como parâmetro
JMP	endereço	desviar para a instrução do <b>endereço</b> passado como parâmetro
JMT	endereço	desviar quando for verdadeiro para a instrução do <b>endereço</b> passado como parâmetro
STP	0	finalizar a execução
entrada / saída de dados		
REA	tipo	ler dados do <b>tipo</b> passado como parâmetro, sendo que <b>1</b> indica um dado do tipo inteiro, <b>2</b> indica um dado do tipo real, <b>3</b> indica um dado do tipo literal
WRT	0	escrever dados

## instrução: ADD, 0

```

pilha [topo - 1] ← pilha [topo - 1] + pilha [topo]
topo ← topo - 1
ponteiro ← ponteiro + 1

```

## instrução: ALB, deslocamento

```

PARA i DE (topo + 1) ATÉ (topo + deslocamento)
FAÇA pilha [i] := FALSE
FIMPARA
topo ← topo + deslocamento
ponteiro ← ponteiro + 1

```

## instrução: ALI, deslocamento

```

PARA i DE (topo + 1) ATÉ (topo + deslocamento)
FAÇA pilha [i] := 0
FIMPARA
topo ← topo + deslocamento
ponteiro ← ponteiro + 1

```

## instrução: ALR, deslocamento

```

PARA i DE (topo + 1) ATÉ (topo + deslocamento)
FAÇA pilha [i] := 0.0
FIMPARA
topo ← topo + deslocamento
ponteiro ← ponteiro + 1

```

## instrução: ALS, deslocamento

```

PARA i DE (topo + 1) ATÉ (topo + deslocamento)
FAÇA pilha [i] := ''
FIMPARA
topo ← topo + deslocamento
ponteiro ← ponteiro + 1

```

## instrução: AND, 0

```

pilha [topo - 1] ← (pilha [topo - 1]) E (pilha [topo])
topo ← topo - 1
ponteiro ← ponteiro + 1

```

instrução: BGE, 0

pilha [topo - 1]  $\leftarrow$  (pilha [topo - 1]  $\geq$  pilha [topo])  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: BGR, 0

pilha [topo - 1]  $\leftarrow$  (pilha [topo - 1]  $>$  pilha [topo])  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: DIF, 0

pilha [topo - 1]  $\leftarrow$  (pilha [topo - 1]  $\neq$  pilha [topo])  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: DIV, 0

SE pilha [topo] = 0  
ENTÃO ESCRIVA ('*RUNTIME error: divisão por 0*')  
HALT  
FIMSE  
pilha [topo - 1]  $\leftarrow$  pilha [topo - 1] / pilha [topo]  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: EQL, 0

pilha [topo - 1]  $\leftarrow$  (pilha [topo - 1] = pilha [topo])  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: JMF, endereço

SE (pilha [topo] = FALSE)  
ENTÃO ponteiro  $\leftarrow$  endereço  
SENÃO ponteiro  $\leftarrow$  ponteiro + 1  
FIMSE  
topo  $\leftarrow$  topo - 1

instrução: JMP, endereço

ponteiro  $\leftarrow$  endereço

instrução: JMT, endereço

SE (pilha [topo] = TRUE)  
ENTÃO ponteiro  $\leftarrow$  endereço  
SENÃO ponteiro  $\leftarrow$  ponteiro + 1  
FIMSE  
topo  $\leftarrow$  topo - 1

instrução: LDV, endereço

topo  $\leftarrow$  topo + 1  
pilha [topo]  $\leftarrow$  pilha [endereço]  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: LDB, constante

topo  $\leftarrow$  topo + 1  
pilha [topo]  $\leftarrow$  constante  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: LDI, constante

topo  $\leftarrow$  topo + 1  
pilha [topo]  $\leftarrow$  constante  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: LDR, constante

topo  $\leftarrow$  topo + 1  
pilha [topo]  $\leftarrow$  constante  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: LDS, constante

topo  $\leftarrow$  topo + 1  
pilha [topo]  $\leftarrow$  constante  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: MUL, 0

pilha [topo-1]  $\leftarrow$  pilha [topo-1] \* pilha [topo]  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: NOT, 0

pilha [topo]  $\leftarrow$  NAO (pilha [topo])  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: OR, 0

pilha [topo - 1]  $\leftarrow$  (pilha [topo - 1]) OU (pilha [topo])  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: REA, tipo

topo  $\leftarrow$  topo + 1  
LEIA (pilha [topo])  
ponteiro  $\leftarrow$  ponteiro + 1  
SE tipo (pilha [topo])  $\neq$  tipo  
ENTÃO ESCRIVA ('*RUNTIME error*')  
HALT  
FIMSE

instrução: SME, 0

pilha [topo - 1]  $\leftarrow$  (pilha [topo - 1]  $\leq$  pilha [topo])  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: SMR, 0

pilha [topo - 1]  $\leftarrow$  (pilha [topo - 1]  $<$  pilha [topo])  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: STR, endereço

pilha [endereço]  $\leftarrow$  pilha [topo]  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: STP, 0

HALT

instrução: SUB, 0

pilha [topo-1]  $\leftarrow$  pilha [topo-1] - pilha [topo]  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1

instrução: WRT, 0

ESCREVA (pilha [topo])  
topo  $\leftarrow$  topo - 1  
ponteiro  $\leftarrow$  ponteiro + 1