Instruções que manipulam vetores

- **Copia um *vetores* de uma posição para outra**
- **Procura um determinado byte ou uma determinada word em um *vetor***
- **Armazena um caractere em um *vetor***
- **Compara vetores de caracteres alfabéticos**

INSTRUÇÃO	DESTINO	ORIGEM	COM BYTE	COM WORD
MOVE VETOR	ES:DI	DS:SI	MOVSB	MOVSW
COMPARA VETOR	ES:DI	DS:SI	CMPSB	CMPSW
ARMAZENA VETOR	ES:DI	AL ou AX	STOSB	STOSW
CARREGA VETOR	AL ou AX	DS:SI	LODSB	LODSW
PROCURA VETOR	ES:DI	AL ou AX	SCASB	SCASW

Flag de Direção - DF

- DF = 0, SI e DI são incrementados
- DF = 1, SI e DI são decrementados

STRING1 DB 'ABCDE'

Endereço	Conteúdo	Caractere ASCII
0200h	041h	A
0201h	042h	B
0202h	043h	C
0203h	044h	D
0204h	045h	E

Instruções CLD e STD

- **Para fazer DF = 0**

```
CLD    ; zera DF
```

- **Para fazer DF = 1**

```
STD    ; seta DF
```

- **OBS – CLD e STD não afetam outros outros FLAGS**

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Movendo um vetor

MOVSB

```
.....
.DATA
        STR1 DB 'OSC 2S21'
        STR2 DB  8 DUP(?)
        QTDE EQU 8
.CODE
        MOV AX,@DATA
        MOV DS,AX           ; inicializa DS
        MOV ES,AX           ; inicializa ES
        CLD
        LEA   SI,STR1       ; origem do string
        LEA   DI,STR2       ; destino do string
        CLD                 ; zera DF
        MOV   CX,QTDE
REPETE:
        MOVSB     ; move da origem para o destino
        LOOP  REPETE
        .........
```

Diretiva REP

```
.....
.DATA
        STR1 DB 'OSC 2S21'
        STR2 DB  8 DUP(?)
        QTDE EQU 8
.CODE
      MOV AX,@DATA
       MOV DS,AX            ; inicializa DS
       MOV ES,AX            ; inicializa ES
       CLD
       LEA   SI,STR1        ; origem do string
       LEA   DI,STR2        ; destino do string
       CLD                  ; zera DF
       MOV   CX,QTDE
       REP MOVSB      ; move da origem para o destino
```

- **MOVSB – NÃO AFETA OS FLAGS**

- **PROGRAMA QUE MOVE UM VETOR DE CARACTERES EM STR1 PARA STR2**

```
.MODEL SMALL
.DATA
        STR1 DB 'OSC 2S21'
        STR2 DB 8 DUP(?),'$'
        TAM EQU 8
.CODE
MAIN PROC
        MOV AX,@DATA
        MOV DS, AX
        MOV ES, AX
        MOV CX,8
        LEA SI,STR1
        LEA DI,STR2
        CLD
_LOOP
        REP MOVSB
        LOOP _LOOP
        MOV AH,9
        LEA DX,STR2
        INT 21H
        MOV AH,4CH
        INT 21H
MAIN END
END MAIN
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **Exercícios**

Escreva um programa que copia um vetor de uma origem para um destino, de forma inversa

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

```
 .....
.DATA
                STR1 DB 'OSC 2S21'
                STR2 DB  8 DUP(?)
                QTDE EQU 8
.CODE
          MOV AX,@DATA
           MOV DS,AX              ; inicializa DS
           MOV ES,AX              ; inicializa ES
           LEA    SI,STR1+7        ; origem do string
           LEA    DI,STR2          ; destino do string
           STD                     ; seta DF
           MOV    CX,QTDE
MOVE:
            MOVSB
            ADD DI,2
            LOOP MOVE
.....
```

- **Exercícios**

Escreva um programa que insira um word (3) em um vetor, depois do 2.

ARR DW 1,2,4,5,6,?

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Inserindo um elemento em um vetor de word

MOVSW

```
.....
.DATA
        ARR DW 1,2,4,5,6,?
        .CODE
        MOV AX,@DATA
        MOV DS,AX           ; inicializa DS
        MOV ES,AX           ; inicializa ES
        STD                 ; seta DF
        LEA   SI,ARR+8h     ; aponta para 6
        LEA   DI,ARR+0Ah    ; aponta para ?
        MOV   CX,3
        REP   MOVSW    ; move 4,5,6 para o fim
        MOV WORD PTR [DI],3
        .........
```

- **MOVSW – NÃO AFETA OS FLAGS**

STOSB/STOSW – Armazena byte/word em um vetor

- MOVE O CONTEÚDO DE AL/AX PARA A POSIÇÃO ES:DI. DI É INCEMENTADO/DECREMENTADO

```
.....
.DATA
        STR1 DB 'OSC 2S21'
        .CODE
        MOV AX,@DATA
        MOV ES,AX           ; inicializa ES
        CLD                 ; zera DF
        LEA   DI,STR1+3     ; aponta STR1+3
        MOV AL, 'X'
        STOSB         ; 'OSCX2S21'
        .........
```

- STOSB/STOSW – NÃO AFETAM OS FLAGS

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

1 - Escreva um procedimento que leia e armazene um vetor, usando STOSB. Fim de leitura = *carriage return* (0Dh)

2 – Reescrever levando em consideração o *backspace* (8h).

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

1 - Escreva um procedimento que leia e armazene um vetor, usando STOSB. Fim de leitura = carriage return (0Dh)

```
le2_str proc
; entrada: si aponta para a posição inicial do local de armazenamento do str
; saída:    si aponta para o str
            cx tamanho do str
          push ax
          push dx
          xor cx,cx
          cld
          mov ah,1
le2_car:
          int 21h
          cmp al,0dh
          je fim2
          stosb
          inc cx
          jmp le2_car
fim2:
          pop dx
          pop ax
          ret
le2_str endp
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

2 - Escreva um procedimento que leia e armazene um vetor, usando STOSB. Levar em consideração o backspace.

```
LE_STR PROC NEAR
; entrada   DI = offset do string
; saída :   DI = offset do string
;          BX = número de caracteres

   PUSH AX
   PUSH DI
   CLD
   XOR BX,BX
   MOV AH,1
   INT 21H
_WHILE1:
   CMP AL, 0Dh
   JE FIM_WHILE1
   CMP AL,8h    ; backspace
   JNE _ELSE1
   DEC DI
   DEC BX
   JMP _LE
_ELSE1:
   STOSB
   INC BX
_LE:
   INT 21h
   JMP _WHILE1
FIM_WHILE1:
   POP DI
   POP AX
   RET
   LE_STR ENDP
```

LODSB/LODSW – Carrega um byte/word de um vetor em AL/AX

- MOVE O CONTEÚDO DA POSIÇÃO DS:SI PARA AL/AX PARA. SI É INCEMENTADO/DECREMENTADO

```
.....
.DATA
        STR1 DB 'OSC 2S21'
        .CODE
        MOV AX,@DATA
        MOV DS,AX           ; inicializa ES
        CLD                 ; zera DF
        LEA   SI,STR1    ; aponta STR1
        LODSB               ; AL = 'O'
        ..........
```

- LODSB/LODSW – NÃO AFETAM OS FLAGS

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Escreva um procedimento que imprima um vetor de caracteres, usando LODSB

```
IMP2  PROC
        MOV AH,1
        LEA SI,STR
        MOV CX, NUM
IMPRIME2:
        LODSB
        MOV DL, AL
        INT 21H
        LOOP IMPRIME2
        RET
IMP2 ENDP
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Escreva um procedimento que imprima um vetor de caracteres, usando LODSB

```
IMP_STR PROC NEAR
; entrada SI = offset do string
;        BX = número de caracteres
; saída : nenhuma
   PUSH AX
   PUSH BX
   PUSH CX
   PUSH DX
   PUSH SI
   MOV CX,BX
   JCXZ  SAIDA
   CLD
   MOV AH,2
LACO:
   LODSB
   MOV DL,AL
   INT 21H
   LOOP LACO
SAIDA:
   POP SI
   POP DX
   POP CX
   POP BX
   POP AX
   RET
IMP_STR ENDP
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

SCASB/SCASW – Procura um byte/word em um vetor

- **PROCURA O CONTEÚDO DE AL/AX EM UM STRING/VETOR ES:DI. DI É INCEMENTADO/DECREMENTADO. RESULTADO EM ZF**

```
.....
.DATA
        STR1 DB 'OSC 2S21'
        .CODE
        MOV AX,@DATA
        MOV ES,AX           ; inicializa ES
        CLD                 ; zera DF
        LEA   DI,STR1       ; aponta STR1
        MOV AL, 'S'
        SCASB               ; so primeiro byte
                .....
```

```
.....
.DATA
        STR1 DB 'OSC 2S21'
        .CODE
        MOV AX,@DATA
        MOV ES,AX           ; inicializa ES
        MOV CX, 8
        CLD                 ; zera DF
        LEA   DI,STR1       ; aponta STR1
        MOV AL, 'S'
        REPNE SCASB         ; repete enquanto não achar
                .....
```

- **SCASB/SCASW –AFETAM APENAS O ZF**

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

CMPSB/CMPSW – Compara 2 vetores

- **COMPRA 2 STRING/VETOR. VETOR ORIGEM EM DS:SI E DESTINO EM ES:DI. SI/DI É INCEMENTADO/DECREMENTADO. RESULTADO EM ZF**

```
.....
.DATA
        STR1 DB 'ABC'
        STR2 DB 'ADC'
        .CODE
        MOV AX,@DATA
        MOV DS,AX
        MOV ES,AX           ; inicializa ES
        CLD                 ; zera DF
        LEA SI,STR1         ; aponta STR1
        LEA DI, STR2
        CMPSB
        CMPSB
        ..........
```

- **CMPSB/CMPSW –AFETAM OS FLAGS ZF E SF**

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

CMPSB/CMPSW – Compara 2 vetores

- **COMPRA 2 STRING/VETOR. VETOR ORIGEM EM DS:SI E DESTINO EM ES:DI. SI/DI É INCEMENTADO/DECREMENTADO. RESULTADO EM ZF**

```
.....
.DATA
        STR1 DB 'ABC'
        STR2 DB 'ADC'
        .CODE
        MOV AX,@DATA
        MOV DS,AX
        MOV ES,AX          ; inicializa ES
        CLD                ; zera DF
        LEA SI,STR1        ; aponta STR1
        LEA DI, STR2
        CMPSB
        CMPSB
        .............
```

- **CMPSB/CMPSW –AFETAM OS FLAGS ZF E SF**

```
.DATA
        STR1 DB 'ABC'
        STR2 DB 'ADC'
        NAO DB 'NAO$'
        SIM DB 'SIM$'
        .CODE
        MOV AX,@DATA
        MOV DS,AX
        MOV ES,AX          ; inicializa ES
        MOV CX,3
        CLD                ; zera DF
        LEA SI,STR1        ; aponta STR1
        LEA DI,STR2
        REPE CMPSB
         JZ SIM
        LEA DX, NAO
        JMP IMPR
SIM:
        LEA DX,SIM
IMPR:
        MOV AH,9
        INT 21H
        ........
```