




# Programação de Software Básico – Linguagem Assembly – Comando Loop e Manipulação de Strings

MATA49

Prof. Babacar Mane

2021.1 – Aula 5



# Programação de Software Básico

## loop e manipulação de strings

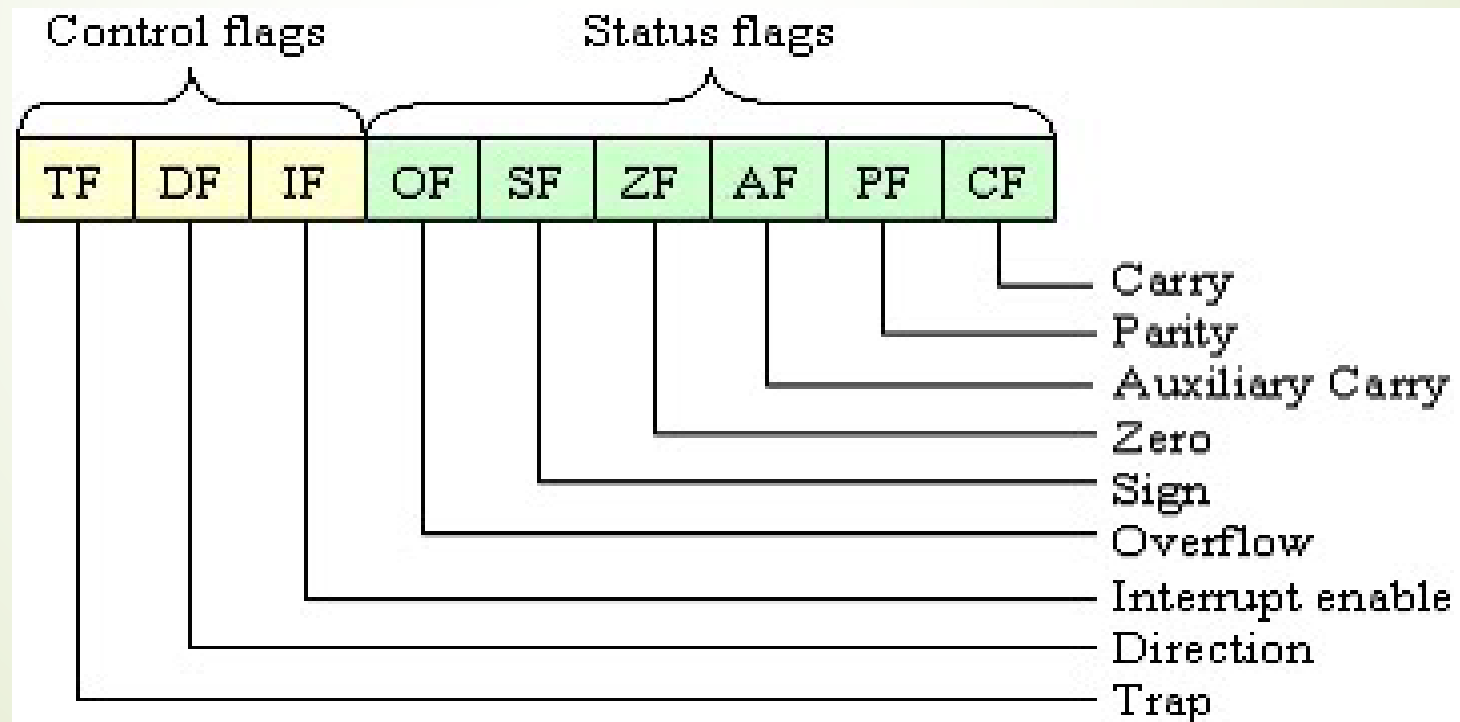
### Conteúdo

- **Loop condicional**
- **Manipulação de Strings**

# Programação de Software Básico

## Vetores

- Eflags - ZF e DF





# Programação de Software Básico

## Vetores

- Eflags - ZF e DF
- **Zero flag (ZF)**: Indica se uma operação lógica ou aritmética gera um resultado zero
- **Direction Flag (DF)** - usado por algumas instruções para processar cadeias de dados.



# Programação de Software Básico

## Comando Loop condicional

- Instrução de repetição que mantem a contagem no registrador CX
  - Loop
  - Loopz e loope
  - Loopnz e loopne



# Programação de Software Básico

## Comando Loop condicional

- Instrução de repetição que mantém a contagem no registrador CX
- Sintaxe:
  - Loop *destino*
- Lógica:
  - $ECX \leftarrow ECX - 1$
  - Se  $CX \neq 0$  salta para o destino



# Programação de Software Básico

## Comando Loop condicional

- Instrução de repetição que mantém a contagem no registrador CX
- Sintaxe:
  - *Loope destino*
  - *Loopz destino*
- Lógica:
  - *ECX <- ECX - 1*
  - *Se ECX > 0 e ZF = 1, salta para o destino*
- *Útil quando rastreia um vetor para encontrar o primeiro elemento que não coincide com um dado valor*





# Programação de Software Básico

## Comando Loop condicional

- Instrução de repetição que mantém a contagem no registrador CX
- Sintaxe:
  - `Loopnz destino`
  - `Loopne destino`
- Lógica:
  - `ECX <- ECX - 1`
  - Se `ECX > 0` e `ZF = 0`, salta para o destino
- Útil quando se rastreia um vetor em busca do primeiro elemento que coincide com um dado valor



# Programação de Software Básico

## Comando manipulação de strings

- Instruções de manipulação de Strings

### Registradores implícitos

- [E]SI      índice para string fonte
- [E]DI      índice para string destino
- ES          segmento do string destino
- [E]CX      contador
- AL/AX/EAX valor de busca (destino p/ LODS, fonte p/ STOS)
- DF          0 (auto incremento p/ DI, SI)  
                1 (auto decremento p/ DI, SI)
- ZF          condição de término para busca e comparação



# Programação de Software Básico

## Comando manipulação de strings

- Manipulação de Strings
- Os registradores ESI e EDI são usados como ponteiros para endereços de memória
- Em instruções de manipulação de dados na memória eles possuem a seguinte semântica:
  - **ESI : fonte (Source)**
  - **EDI : destino (Destination)**



# Programação de Software Básico

## Comando manipulação de strings

- Manipulação de Strings
- Os valores dos registradores ESI e EDI são incrementados ou decrementados em função do flag DF
  - Se  $DF = 0$  então os valores dos ESI e EDI são incrementados
  - Se  $DF = 1$  então os valores dos ESI e EDI são decrementados
- Instruções:
  - CLD: Zera o Direction Flag (DF)
  - STD: Atribui 1 ao Direction Flag (DF)



# Programação de Software Básico

## Comando manipulação de strings

- Instruções de manipulação de Strings
  - MOVS: move source string to destination string
  - CMPS: compare source string with destination string
  - SCAS: scan destination string
  - LOADS: load into AL/AX/EAX from source string
  - STOS: store AL/AX/EAX into destination string

# Programação de Software Básico

## Comando manipulação de strings

- Instruções de manipulação de Strings
  - MOVS: move source string to destination string

Copia dados da memória apontado pelo registrador ESI para a memória indexada pelo registrador EDI

|       |                         |
|-------|-------------------------|
| MOVSB | Move (copy) bytes       |
| MOVSW | Move (copy) words       |
| MOVSD | Move (copy) doublewords |

| Instruction | Value Added or Subtracted from ESI and EDI |
|-------------|--|
| MOVSB       | 1  |
| MOVSW       | 2  |
| MOVSD       | 4  |

# Programação de Software Básico

## Comando manipulação de strings

- Exemplo:
  - msg1 db "Hello World"
  - msg2 db "AB"
  - mov esi,msg2 ;fonte
  - mov edi , msg1 ;destino
  - cld
  - movsb ;msg1 = "Hello World"



# Programação de Software Básico

## Comando manipulação de strings

- Exemplo:

```
array1 dd 1,2,3,4,5
```

```
array2 dd 6,7,8,9,10
```

```
mov esi, array1
```

```
mov edi , array2
```

```
mov ecx, 5
```

```
cld;
```

```
mark:
```

```
    movsd
```

```
    ;array2 = 1,2,3,4,5
```

```
loop mark
```





# Programação de Software Básico

## Comando Repetição

- Comando de repetição: REP
- Repete uma instrução um número específico de vezes
  - O ECX contabiliza o número de repetições e é decrementado a cada execução do REP (similar a instrução LOOP)
  - Sintaxe: REP <instrução>

# Programação de Software Básico

## Comando Repetição

- Comando de repetição: REP

O exemplo anterior pode ser escrito:

```
array1 dd 1,2,3,4,5
```

```
array2 dd 6,7,8,9,10
```

```
mov esi, array1
```

```
mov edi , array2
```

```
mov ecx, 5
```

```
rep movsd      ;array2 = 1,2,3,4,5
```

# Programação de Software Básico

## Comando manipulação de strings

- Instruções de manipulação de Strings
  - CMPS: compare source string with destination string
  - Compara as memória indexadas pelos registradores ESI e EDI
    - *Modifica os mesmo bits que a instrução CMP*
    - *Incrementa/decrementa bytes (depende da instrução) nos registradores ESI e EDI*
  - Incrementar => DF = 0 ; Decrementar => DF = 1

|       |                     |
|-------|---------------------|
| CMPSB | Compare bytes       |
| CMPSW | Compare words       |
| CMPSD | Compare doublewords |

# Programação de Software Básico

## Comando manipulação de strings

- Instruções de manipulação de Strings: CMPS
- Exemplo: Comparação de dois blocos de memória

```
segment .text
    cld
    mov     esi, block1        ; address of first block
    mov     edi, block2        ; address of second block
    mov     ecx, size          ; size of blocks in bytes
    repe    cmpsb              ; repeat while Z flag is set
    je      equal              ; if Z set, blocks equal
    ; code to perform if blocks are not equal
    jmp     onward
equal:
    ; code to perform if equal
onward:
```



# Programação de Software Básico

## Comando manipulação de strings

- Instruções de manipulação de Strings: CMPS
  - Compara os registradores AL/AX/EAX com a memória apontada pelo registrador EDI
    - Modifica os mesmo bits que a instrução CMP
    - Incrementa/decrementa bytes (depende da instrução) no registrador EDI
- Incrementar => DF = 0 ; Decrementar => DF



# Programação de Software Básico

## Comando manipulação de strings

- Instruções de manipulação de Strings
  - SCAS: scan destination string
    - SCASB (bytes)
    - SCASW (words)
    - SCASD (DoubleWords)



# Programação de Software Básico

## Comando manipulação de strings

- Instruções de manipulação de Strings: SCAS

```
segment .bss
array          resd 100

segment .text
    cld
    mov     edi, array      ; pointer to start of array
    mov     ecx, 100        ; number of elements
    mov     eax, 12         ; number to scan for
lp:
    scasd
    je      found
    loop    lp
    ; code to perform if not found
    jmp     onward
found:
    sub     edi, 4           ; edi now points to 12 in array
    ; code to perform if found
onward:
```





# Programação de Software Básico

## Comando manipulação de strings

- Instruções de manipulação de Strings: STOS
- STOS: store AL/AX/EAX into destination string
- Escreve na memória indexada pelo registrador EDI o conteúdo armazenado no registrador AL/AX/EAX
  - Incrementa/decrementa bytes (depende da instrução) no registrador EDI
- Incrementar => DF = 0 ; Decrementar => DF

# Programação de Software Básico

## Comando manipulação de strings

- Instruções de manipulação de Strings: STOSx

```
segment .data
array1 dd 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

segment .bss
array2 resd 10

segment .text
    cld                                ; don't forget this!
    mov     esi, array1
    mov     edi, array2
    mov     ecx, 10
lp:
    lodsd
    stosd
    loop   lp
```



# Programação de Software Básico

## Comando manipulação de strings

- Instruções de manipulação de Strings
  - **LOADS:** load into AL/AX/EAX from source string
  - Carrega os dados armazenados na memória apontada pelo registrador ESI e armazena em AL/AX/EAX
  - Incrementa/decrementa bytes (depende da instrução) no registrador ESI
  - Incrementar => DF = 0 ; Decrementar => DF

# Programação de Software Básico

## Comando Repetição

- Comando de repetição: REPx

- REP repeat
- REPE / REPZ repeat while equal/repeat while zero
- REPNE / REPNZ repeat while not equal/repeat while not zero
  1. Se  $ECX = 0$ , então não executa a primitiva e encerra a repetição
  2. Se  $ECX > 0$ , então:
    - 2.1 Executa a operação primitiva, atualizando os flags
    - 2.2 Decrementa ECX, sem afetar os flags
    - 2.3 Volta para a repetição, de acordo com o prefixo:
      - 2.3.1 Se for REP, repete incondicionalmente
      - 2.3.2 Se for REPE/Z, somente repete se  $ZF=1$
      - 2.3.3 Se for REPNE/NZ, somente repete se  $ZF=0$