

Linguagem de Montagem

Sistemas Computacionais — Prática

Ciência de Dados e Inteligência Artificial

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

Estrutura de um Programa

- **Stack segment (.STACK)**
 - Armazenamento temporário de endereços e dados.
- **Data segment (.DATA)**
 - Declaração de variáveis.
 - Cada variável é assinada a um espaço de memória.
 - Exemplo 1: `A DW 2` inicializa a variável `A` com o valor `2`
 - Exemplo 2: `SUM DW ?` cria a variável `SUM` sem inicializá-la.
- **Code segment (.CODE)**
 - Contém as instruções do programa (procedures).
 - Exemplo: `MAIN PROC ... MAIN ENDP .`

Programa Simples: Hello World

```
TITLE PGM1: HELLO WORLD
.MODEL SMALL
.STACK 100h
.DATA
    msg db 'Hello, World!$', 0
.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX
    MOV DX, OFFSET msg
    MOV AH, 09h
    INT 21h
    MOV AX, 4C00h
    INT 21h
MAIN ENDP
END MAIN
```

Explicação do Programa Hello World

- `.MODEL SMALL`
 - Define o modelo de memória como SMALL.
 - No modelo SMALL, o código e os dados cabem em um único segmento de 64 KB.
- `.STACK 100h`
 - Reserva 256 bytes (100h em hexadecimal) de espaço para a pilha.
 - A pilha é usada para armazenar endereços de retorno e variáveis temporárias.
- `msg db 'Hello, World!$', 0`
 - Define uma variável msg que contém a string `'Hello, World!$'`
 - O caractere `$` no final é obrigatório para que a interrupção `INT 21h`, função `09h` saiba onde parar a impressão.

Explicação do Programa Hello World (cont.)

- `MAIN PROC` : Define o início do procedimento `MAIN` .
- `MOV AX, @DATA` : Carrega o endereço do segmento de dados (`@DATA`) no registrador AX.
- `MOV DS, AX` : Define DS (Data Segment) para apontar para o segmento de dados.
- `MOV DX, OFFSET msg` : Carrega em DX o endereço da string msg.
- `MOV AH, 09h` : Define a função `09h` da interrupção `INT 21h` do DOS, que imprime uma string na tela. O DOS lê o endereço armazenado em `DX` , busca a string e exibe até encontrar o caractere `$` .
- `MAIN ENDP` : Indica o fim do procedimento `MAIN` .
- `END MAIN` : Indica ao assembler que a execução deve começar a partir de `MAIN`
- Finalização do programa com `MOV AX, 4C00h` e `INT 21h` .

Soma de Dois Dígitos

```
TITLE PGM1: SOMA PROGRAM
.MODEL SMALL
.STACK 100h
.DATA
    msg1 db 'Resultado da soma: $' ; Mensagem inicial
    resultado db '0', '$' ; Local para armazenar o resultado em ASCII
.CODE
MAIN PROC
    ; Inicializa o segmento de dados
    MOV AX, @DATA
    MOV DS, AX
    ; Exibe a mensagem inicial
    MOV DX, OFFSET msg1
    MOV AH, 09h
    INT 21h
    ; Realiza a soma: 5 + 3
    MOV AL, 5 ; Primeiro número
    ADD AL, 3 ; Soma com o segundo número
    ; Converte o resultado para ASCII (somando 48)
    ADD AL, 48
    MOV resultado, AL
    ; Exibe o resultado
    MOV DX, OFFSET resultado
    MOV AH, 09h
    INT 21h
    ; Finaliza o programa
    MOV AX, 4C00h
    INT 21h
MAIN ENDP
END MAIN
```

Explicação do Programa Soma

Linha	Descrição
<code>MOV AX, @DATA / MOV DS, AX</code>	Inicializa o segmento de dados.
<code>MOV DX, OFFSET msg1 / INT 21h</code>	Exibe "Resultado da soma: " .
<code>MOV AL, 5</code>	Armazena o número 5 em AL .
<code>ADD AL, 3</code>	Soma 3 ao valor de AL .
<code>ADD AL, 48</code>	Converte o número para ASCII.
<code>MOV resultado, AL</code>	Armazena o número convertido na variável resultado .
<code>MOV DX, OFFSET resultado / INT 21h</code>	Exibe o número na tela.
<code>MOV AX, 4C00h / INT 21h</code>	Finaliza o programa.

Programa Soma Modificado (lê dois dígitos)

```
.MODEL SMALL
.STACK 100h
.DATA
    resultado db '0', '$' ; Armazena o resultado em ASCII
.CODE
MAIN PROC
    ; Inicializa o segmento de dados
    MOV AX, @DATA
    MOV DS, AX
    ; Lê o primeiro número
    MOV AH, 01h
    INT 21h
    SUB AL, 48 ; Converte ASCII para número real (0-9)
    MOV BL, AL ; Armazena o primeiro número em BL
    ; Lê o segundo número
    MOV AH, 01h
    INT 21h
    SUB AL, 48 ; Converte ASCII para número real (0-9)
    ADD BL, AL ; Soma os dois números
    ; Converte o resultado para ASCII
    ADD BL, 48
    MOV resultado, BL ; Armazena o valor convertido
    ; Exibe apenas o resultado
    MOV DX, OFFSET resultado
    MOV AH, 09h
    INT 21h
    ; Finaliza o programa
    MOV AX, 4C00h
    INT 21h
MAIN ENDP
END MAIN
```


Exercícios

1. Modifique o programa para subtrair dois números ao invés de somar.
2. Adapte o código para somar dois números digitados pelo usuário.

Programa Contador de 1 a 5

```
.MODEL SMALL
.STACK 100h
.DATA
    msg db 'Contando: $'    ; Mensagem inicial
    newline db 0Dh, 0Ah, '$' ; Nova linha
.CODE
MAIN PROC
    ; Inicializa o segmento de dados
    MOV AX, @DATA
    MOV DS, AX
    ; Exibe a mensagem inicial
    MOV DX, OFFSET msg
    MOV AH, 09h
    INT 21h
    ; Inicializa o contador (de 1 a 5)
    MOV CX, 5    ; Loop de 5 iterações
    MOV AL, '1' ; Inicia com o caractere '1'
loop_inicio:
    ; Exibe o número atual
    MOV DL, AL
    MOV AH, 02h
    INT 21h
    ; Adiciona uma nova linha
    MOV DX, OFFSET newline
    MOV AH, 09h
    INT 21h
    ; Incrementa o número (próximo caractere ASCII)
    INC AL
    ; Decrementa o contador e repete se CX > 0
    LOOP loop_inicio
    ; Finaliza o programa
    MOV AX, 4C00h
    INT 21h
MAIN ENDP
END MAIN
```

Explicação do Programa Contador de 1 a 5

Linha	Explicação
<code>MOV AX, @DATA / MOV DS, AX</code>	Inicializa o segmento de dados.
<code>MOV DX, OFFSET msg / INT 21h</code>	Exibe "Contando: " .
<code>MOV CX, 5</code>	Define o loop para repetir 5 vezes.
<code>MOV AL, '1'</code>	Define o primeiro número ASCII ('1').
<code>MOV DL, AL / INT 21h</code>	Exibe o número na tela.
<code>MOV DX, OFFSET newline / INT 21h</code>	Exibe uma nova linha.
<code>INC AL</code>	Incrementa o número (ASCII '1' → '2' , etc.).
<code>LOOP loop_inicio</code>	Decrementa CX e repete enquanto CX > 0 .
<code>MOV AX, 4C00h / INT 21h</code>	Finaliza o programa.

Programa Echo

```
TITLE PGM1: ECHO PROGRAM
.MODEL small
.STACK 100h
.DATA
.CODE
MAIN PROC
    MOV AH,2
    MOV DL,'?'
    INT 21H
    MOV AH,1
    INT 21H
    MOV BL,AL
    MOV AH,2
    MOV DL,0DH
    INT 21H
    MOV DL,0AH
    INT 21H
    MOV DL,BL
    INT 21H
    MOV AH,4CH
    INT 21H
MAIN ENDP
    END MAIN
```

Exercícios

1. Modifique o programa para contar de 5 a 1 (ordem decrescente).
2. Altere o código para exibir os números de 1 a 9.
3. Modifique para imprimir letras de A a E ao invés de números.