

**Universidade de São Paulo**

Organização de Computadores

Dr. Jorge Luiz e Silva

Cap 8

# Subrotinas e Parâmetros

Seqüência de instruções repetitivas.

Na chamada existe o DESVIO para a rotina

No retorno existe o RETORNO para o ponto em que foi desviado

Chamada      CALL NOME

              Sistema armazena o endereço de retorno na pilha

Retorno      RET

              o sistema faz retornar ao endereço armazenado na pilha

# Definição de Procedure

NOME            PROC    NEAR (uma palavra na pilha)  
                                  FAR    (duas palavras na pilha)

I

I

I

RET

NOME            ENDP

# Parâmetros

Transferência através de registradores

Posição de memória

Pilha etc.

Pilha → LIFO (last in first out)

# Uma rotina de E/S

MULTIO

PROC

NEAR

- ; AH → função
- ; 1 → leitura com eco e dado em AL
- ; 2 → escrita de caracter em DL
- ; 9 → escrita de string posicionando em DX
- ; terminando por \$
- ;

# Rotina E/S cont...

```
CMP AH,1  
JZ ECO  
CMP AH,2  
JZ ESCRE  
CMP AH,9  
JZ STRING  
RET
```

# Rotina E/S cont...

;

; Iniciando tratamento de ECO

;

ECO:           MOV  DX,STATUS

M1:            IN     AL,DX

              TEST  AL,2

              JZ     M1

              MOV  DX,DADO

              IN     AL,DX

              MOV  AH,AL

              MOV  DX,STATUS

# Rotina E/S cont...

```
M2:      IN     AL,DX
          TEST  AL,1
          JZ   M2
          MOV  DX,DADO
          MOV  AL,AH
          OUT  DX,AL
          RET
```



# Rotina E/S cont...

;  
;       iniciando tratamento de escrev  
;

```
ESCRE:       MOV  AH,DL  
              MOV  DX,STATUS  
M3:           IN   AL,DX  
              TEST AL,1  
              JZ   M3  
              MOV  DX,DADO  
              MOV  AL,AH  
              OUT  DX,AL  
              RET
```

# Rotina E/S cont...

;

; iniciando tratamento de string

;

STRING: MOV BX,DX

M6: MOV DX,STATUS

M4: IN AL,DX

TEST AL,1

JZ M4

MOV AL,[BX]

CMP AL,'\$'

JZ M5

# Rotina E/S cont...

```
                MOV  DX,DADO
                OUT  DX,AL
                INC  BX
                JMP  M6
M5:             RET
MULTIO         ENDP
```

# Programa Principal

```
DADOS      SEGMENT
  MEN DB 'ESTOU ENVIANDO MENSAGEM:',
  0DH,0AH,'$'
DADOS      ENDS
CODE       SEGMENT
          ASSUME     CS:CODE, DS:DADOS
MULTIO     PROC NEAR
          I
          I
          I
MULTIO     ENDP
```

# Programa Principal cont...

```
INIC: MOV  DX,DADOS
      MOV  DS,DX
      MOV  AH,9
      MOV  DX,OFFSET MEN
      CALL MULTIO
; JUMP PARA O SISTEMA OPERACIONAL
      JMP  SO
CODE  ENDS
      END INIC
```

# Máquinas da linha IBM/PC

INT n° → tipo particular de rotina cuja diferença com Procedure é que além do endereço de retorno, automaticamente o FLAG é salvo na pilha.

INT 21H → rotina para tratar funções do DOS  
relacionados com I/O  
INT 10H → rotina para tratar vídeo

Ex:           MOV AX,2

          INT 10H → apaga a tela toda

Parâmetros da INT 21H

AH ← 1       lê com ECO e dado para AL

AH ← 2       escreve caracter de DI

AH ← 9       escreve string terminada por \$ por DX

AH ← 4Ch     volta para DOS

# Exemplo IBM/PC 1

Programa que pergunte por um dado em um sistema da linha IBM/PC

```
DADOS      SEGMENT

CR         EQU    0DH
LF         EQU    0AH
MEN        DB     'ENVIE SEU DADO:', CR,LF,'$'
X          DB     ?

DADOS      ENDS
```

# Exemplo IBM/PC 1 cont...

CODE SEGMENT

ASSUME CS:CODE, DS:DADOS

INIC: MOV DX,DADOS

MOV DS,DX

; apagando a tela

MOV AX,2

INT 10H

; escrita da mensagem

MOV AH,9

MOV DX,OFFSET MEN

INT 21H



# Exemplo IBM/PC 1 cont...

; lendo um caracter com eco

```
MOV AH,1
```

```
INT 21H
```

```
MOV X,AL
```

; voltando para o dos

```
MOV AH,4Ch
```

```
INT 21H
```

```
CODE ENDS
```

```
END INIC
```

# Exemplo IBM/PC 2

Dado um vetor com n posições fazer um programa que leia esse vetor (8 bits)

```
DADOS      SEGMENT
CR         EQU  0DH
LF         EQU  0AH
MEM0       DB   'DIGITE O VALOR N', CR,LF,'$'
MEN1       DB   'DIGITE O VETOR', CR,LF,'$'
N          DB   ?
VET        DB   10 DUP(?)
DADOS      ENDS
```

# Exemplo IBM/PC 2 cont...

CODE SEGMENT

ASSUME CS:CODE, DS:DADOS

INIC:

MOV DX,DADOS

MOV DS,DX

; apagando a tela

MOV AX,2

INT 10H

; escrita da mensagem para n

MOV AH,9

MOV DX,OFFSET MEN0

INT 21H

# Exemplo IBM/PC 2 cont...

; lendo n com eco

```
MOV AH,1  
INT 21H  
AND AL,0FH  
MOV N,AL
```

; apagando a tela

```
MOV AX,2  
INT 10H
```

; escrita da mensagem para vetor

```
MOV AH,9  
MOV DX,OFFSET MEN1  
INT 21H
```

# Exemplo IBM/PC 2 cont...

; lendo o vetor

```
MOV CL,N
XOR CH,CH
XOR SI,SI
MOV AH,1
INT 21H
AND AL,0FH
MOV VET[SI],AL
INC SI
LOOP T1
```

# Exemplo IBM/PC 2 cont...

; voltando para o dos

```
MOV AH,4Ch
```

```
INT 21H
```

```
CODE
```

```
ENDS
```

```
END INIC
```

# Exemplo IBM/PC 3

Fazer um programa que compare duas variáveis e coloque o maior delas em uma terceira variável.

```
DADOS          SEGMENT
CR             EQU          0DH
LF            EQU          0AH
MEN1 DB       'DIGITE X - ',CR,LF,'$'
MEN2 DB       'DIGITE Y - ',CR,LF,'$'
MEN3 DB       'MAIOR - ',CR,LF,'$'
X             DB           ?
Y             DB           ?
MAIOR        DB           ?
DADOS          ENDS
```

# Exemplo IBM/PC 3 cont...

CODE SEGMENT

ASSUME CS:CODE, DS:DADOS

INIC:

MOV DX,DADOS

MOV DS,DX

; apagando a tela

MOV AX,2

INT 10H

; escrita da mensagem para x

MOV AH,9

MOV DX,OFFSET MEN1

INT 21H



# Exemplo IBM/PC 3 cont...

; lendo x com eco

```
MOV AH,1
INT 21H
AND AL,0FH
MOV X,AL
```

; apagando a tela

```
MOV AX,2
INT 10H
```

; escrita de mensagem de y

```
MOV AH,9
MOV DX,OFFSET MEN2
INT 21H
```

# Exemplo IBM/PC 3 cont...

; lendo y com eco

```
MOV AH,1  
INT 21H  
AND AL,0FH  
MOV Y,AL
```

;

; iniciando a comparação

;

```
MOV AL,X  
CMP AL,Y  
JG T2  
MOV AL,Y
```

T2:

```
MOV MAIOR,AL
```

# Exemplo IBM/PC 3 cont...

; apagando a tela

```
MOV AX,2
```

```
INT 10H
```

; escrita da mensagem para maior

```
MOV AH,9
```

```
MOV DX,OFFSET MEN3
```

```
INT 21H
```

;

; escrita do maior

```
MOV DL,MAIOR
```

```
ADD DL,30H
```

```
MOV AH,2
```

```
INT 21H
```

# Exemplo IBM/PC 3 cont...

; voltando para o dos

```
MOV AH,4Ch
```

```
INT 21H
```

```
CODE
```

```
ENDS
```

```
END INIC
```

# Exemplo Uso de Rotina

## Somar 4 vetores de 8 bits

```
DADOS          SEGMENT
CR    QUE      0DH
LF    QUE      0AH
MEN1  DB       'LER UM VETOR',CR,LF,'$'
MEN2  DB       'DIGITE N - ',CR,LF,'$'
MEN3  DB       'RESULTADO - ',CR,LF,'$'
N     DB       ?
A     VET      20 DUP(?)
B     VET      20 DUP(?)
C     VET      20 DUP(?)
D     VET      20 DUP(?)
E     VET      20 DUP(?)
DADOS          ENDS
```

# Exemplo Uso de Rotina cont...

```
CODE SEGMENT
```

```
ASSUME CS:CODE, DS:DADOS
```

```
LER09      PROC      NEAR
```

```
;rotina que lê valores entre 0-9
```

```
;dado lido fica em DL
```

```
        PUSH AX
```

```
        MOV  AH,1
```

```
        INT  21H
```

```
        AND  AL,0FH
```

```
        MOV  DL,AL
```

```
        POP  AX
```

```
        RET
```

```
LER09      ENDP
```

# Exemplo do Uso de Rotina cont...

```
LERVET      PROC      NEAR
; rotina que lê um vetor de n posições
; usa rotina ler09
;vetor é lido sempre em E
          PUSH SI
          PUSH CX
          PUSH AX
          PUSH DX
; apagando a tela
          MOV  AX,2
          INT  10H
```

# Exemplo do Uso de Rotina cont...

; escrevendo mensagem para ler vetor

```
MOV DX,OFFSET MEN1
```

```
MOV AH,9
```

```
INT 21H
```

; iniciando a leitura da vetor



# Exemplo do Uso de Rotina cont...

```

MOV CL,N
XOR CH,CH
XOR SI,SI
T1: CALL LER09
MOV E[SI],DL
INC SI
LOOP T1
POP DX
POP AX
POP CX
POP SI
RET
LERVET ENDP
```

# Exemplo do Uso de Rotina cont...

INIC:

```
MOV DX,DADOS
```

```
MOV DS,DX
```

; escrita da mensagem para n

;apagando a tela

```
MOV AX,2
```

```
INT 10H
```

```
MOV AH,9
```

```
MOV DX,OFFSET MEN2
```

```
INT 21H
```

# Exemplo do Uso de Rotina cont...

; lendo n com eco

```
CALL LER09
```

```
MOV N,DL
```

; escrita da mensagem vetor a

; apagando a tela

```
MOV AX,2
```

```
INT 10H
```

```
MOV AH,9
```

```
MOV DX,OFFSET MEN2
```

```
INT 21H
```

# Exemplo do Uso de Rotina cont...

;lendo vetor a com eco

```
CALL LERVET
XOR  SI,SI
MOV  CL,N
XOR  CH,CH
T2:  MOV  AL,E[SI]
      MOV  A[SI],AL
      INC  SI
      LOOP T2
```

# Exemplo do Uso de Rotina cont...

; escrita da mensagem vetor b

; apagando a tela

```
MOV    AX,2
INT    10H
MOV    AH,9
MOV    DX,OFFSET MEN2
INT    21H
```

;lendo vetor b com eco

```
CALL   LERVET
XOR    SI,SI
MOV    CL,N
XOR    CH,CH
T3:    MOV    AL,E[SI]
        MOV    B[SI],AL
        INC    SI
        LOOP  T3
```

# Exemplo do Uso de Rotina cont...

; escrita da mensagem vetor c

```
MOV    AX,2
INT    10H
MOV    AH,9
MOV    DX,OFFSET MEN2
INT    21H
```

; lendo vetor c com eco

```
CALL   LERVET
XOR    SI,SI
MOV    CL,N
XOR    CH,CH
T4:    MOV    AL,E[SI]
        MOV    C[SI],AL
        INC    SI
        LOOP  T4
```

# Exemplo do Uso de Rotina cont...

; escrita da mensagem em vetor d

; apagando a tela

```
MOV    AX,2
```

```
INT    10H
```

```
MOV    AH,9
```

```
MOV    DX,OFFSET MEN2
```

```
INT    21H
```

; lendo vetor d com eco

```
CALL   LERVET
```

```
XOR    SI,SI
```

```
MOV    CL,N
```

```
XOR    CH,CH
```

```
T5:    MOV    AL,E[SI]
```

```
MOV    D[SI],AL
```

```
INC    SI
```

```
LOOP   T5
```

# Exemplo do Uso de Rotina cont...

; somando os vetores

```
MOV CL,N
XOR CH,CH
XOR SI,SI
ADD AL,A[SI]
ADD AL,B[SI]
ADD AL,C[SI]
ADD AL,D[SI]
MOV E[SI],AL
INC SI
LOOP T6
```



# Exemplo do Uso de Rotina cont...

;resultados

```
MOV    AX,2
INT    10H
MOV    DX,OFFSET MEN3
MOV    AH,9
INT    21H
```

; escrevendo o vetor resultado

```
      XOR    SI,SI
      MOV    CL,N
      XOR    CH,CH
T7:   MOV    DL,E[SI]
      ADD    DL,30H
      MOV    AH,2
      INT    21H
      INC    SI
      LOOP  T7
```

# Exemplo do Uso de Rotina cont...

;voltando para o dos

MOV AH,4Ch

INT 21H

CODE ENDS

END INIC

# Exercício

**Fazer um programa que leia duas matrizes de ordem  $n$  e calcule o somatório de todos os elementos das primeiras linhas de cada matriz. Coloque o resultado em variáveis diferentes.**

**Obs: leia e mostre os resultados em valores entre (0-9)**