MACRO

- Định nghĩa Macro và gọi Macro
- Vấn đề truyền thông số trong Macro.
- Macro lồng nhau.
- Sử dụng Macro để gọi chương trình con.
- Các toán tử Macro.
- Thư viện Macro
- So sánh việc dùng Macro với Procedure
- Một số Macro mẫu.

ĐỊNH NGHĨA MACRO

- Macro là 1 ký hiệu được gán cho 1 nhóm lệnh ASM – Macro là tên thay thế cho 1 nhóm lệnh.

Tại sao cần có Macro :

- Trong lập trình nhiều lúc ta cần phải viết những lệnh na ná nhau nhiều lần mà ta không muốn viết dưới dạng hàm vì dùng hàm tốn thời gian thực thi, thay vì ta phải viết đầy đủ nhóm lệnh này vào CT, ta chỉ cần viết Macro mà ta đã gán cho chúng.

LÀM QUEN VỚI MACRO

Khi ta có nhiều đoạn code giống nhau, chúng ta có thể dùng macro để thay thế, giống như ta dùng define trong C. Thí dụ chúng ta thay thế đoạn lệnh sau bằng macro để in dấu xuống dòng.

```
MOV DL,13 ; về đầu dòng
MOV AH,2
INT 21H
MOV DL,10 ; xuống dòng mới
MOV AH,2
INT 21H
```

Thay vì phải viết lại 6 dòng lệnh trên, ta có thể tạo 1 macro có tên @Newline để thay thế đoạn code này :

```
@NewLine  Macro
          MOV  DL,13
          MOV  AH,2
          INT  21H
          MOV  DL,10
          MOV  AH,2
          INT  21H
ENDM
```

Sau đó, bất kỳ chỗ nào cần xuống dòng, ta chỉ cần gọi macro @NewLine.

@NewLine

MACRO （tt）

- Khi hợp dịch nội dung nhóm lệnh này mà ta đã gán cho macro sẽ được thay thế vào những nơi có tên macro trước khi CT được hợp dịch thành file OBJ.
  - Ex1 : nhiều khi ta phải viết lại nhiều lần đoạn lệnh xuất ký tự trong DL ra màn hình.
  - MOV AH, 2
  - INT 21H
  - Thay vì phải viết cả 1 cặp lệnh trên mỗi khi cần xuất ký tự trong DL, ta có thể viết Macro PUTCHAR như sau :

```
PUTCHAR MACRO
        MOV AH,2
        INT 21H
ENDM
```

- MỞ RỘNG CỦA MACRO CÓ THỂ XEM TRONG FILE.LIST.
- 3 DIRECTIVE BIÊN DỊCH SAU SẼ QUYẾT ĐỊNH MỞ RỘNG MACRO NHƯ THẾ NÀO.
  - .SALL (SUPRESS ALL) PHẦN MỞ RỘNG MACRO KHÔNG ĐƯỢC IN. SỬ DỤNG KHI MACRO LỚN HAY MACRO ĐƯỢC THAM CHIẾU NHIỀU LẦN TRONG CT.
  - .XALL CHỈ NHỮNG DÒNG MACRO TẠO MÃ NGUỒN MỚI ĐƯỢC IN RA.THÍ DỤ CÁC DÒNG CHÚ THÍCH ĐƯỢC BỎ QUA. ĐÂY LÀ TUỲ CHỌN DEFAULT.
  - .LALL (LIST ALL) TOÀN BỘ CÁC DÒNG TRONG MACRO ĐƯỢC IN RA TRỪ NHỮNG CHÚ THÍCH BẮT ĐẦU BẰNG 2 DẤU ;;

ĐỊNH NGHĨA MACRO

- CÚ PHÁP KHAI BÁO MACRO :

```
MACRO_NAME  MACRO  [ <THÔNG SỐ HÌNH THỨC>]
      STATEMENTS
 ENDM
```

- GỌI MACRO :

```
MACRO_NAME  [<THÔNG SỐ THỰC>, ...]
```

THÔNG SỐ HÌNH THỨC CHỈ CÓ TÁC DỤNG ĐÁNH DẤU VỊ TRÍ CỦA THÔNG SỐ TRONG MACRO. QUAN TRỌNG NHẤT LÀ VỊ TRÍ CÁC THÔNG SỐ.

MACRO TRUYỀN THAM SỐ

```
.MODEL SMALL
.STACK 100H
 PUTCHAR MACRO KT
      MOV DL,KT
      MOV AH,2
      INT 21H
 ENDM
.CODE
  MAIN PROC
    MOV DL, 'A'
    PUTCHAR
   MOV DL, '*"
   PUTCHAR
    MOV AH,4CH
    INT 21H
MAIN ENDP
END MAIN
```

```
SWAP  MACRO BIẾN1, BIẾN2
        MOV AX, BIEN1
        XCHG AX, BIEN2
        MOV BIEN1, AX
 ENDM
```

GỌI : SWAP   TRI1, TRI2

TRAO ĐỔI THAM SỐ CỦA MACRO

MỘT MACRO CÓ THỂ CÓ THÔNG SỐ HOẶC KHÔNG CÓ THÔNG SỐ.

MACRO CÓ THÔNG SỐ

SỬ DỤNG MACRO

```
PUTCHAR   MACRO   CHAR
          MOV  AH,  2
          MOV  DL,  CHAR
          INT  21H
ENDM
```

```
. CODE
.. ...
        PUTCHAR    'A'
        PUTCHAR    'B'
        PUTCHAR    'C'
...
```

MACRO TRUYỀN THÔNG SỐ

Thí dụ : macro @Printstr

Viết chương trình in 2 chuỗi ‘Hello’ và ‘Hi”.

```
.DATA
MSG1 DB ‘Hello’,13,10
MSG2 DB‘Hi’,13,10
.CODE
…….
MOV DX, OFFSET MSG1     ;1
MOV AH,9                ;1
INT 21H                 ;1
MOV DX, OFFSET MSG2     ;2
MOV AH,9                ;2
INT 21H                 ;2
……..
```

Ta thấy đoạn 1 và đoạn 2 gần giống nhau ➔ có thể tạo macro có tham số như sau :

THÍ DỤ VỀ MACRO

```
DISPLAY   MACRO   STRING
          PUSH  AX
          PUSH  DX
        LEA  DX,  STRING
        MOV  AH,9
        INT  21H
          POP  DX
          POP  AX
ENDM
```

GỌI : DISPLAY CHUOI

TRAO ĐỔI THAM SỐ CỦA MACRO

MACRO LOCATE : ĐỊNH VỊ CURSOR MÀN HÌNH

```
LOCATE  MACRO   ROW, COLUMN
        PUSH  AX
        PUSH  BX
        PUSH  DX
        MOV  BX, 0
        MOV  AH, 2
        MOV  DH, ROW
        MOV  DL, COLUMN
        INT  10H
        POP  DX
        POP  BX
        POP  AX
ENDM
```

SỬ DỤNG MACRO

TA CÓ CÁC DẠNG SỬ DỤNG SAU :

LOCATE 10,20

LOCATE ROW, COL

LOCATE CH, CL

CHÚ Ý : KHÔNG DÙNG CÁC THANH GHI AH,AL,BH,BL VÌ SẼ ĐỤNG ĐỘ VỚI CÁC THANH GHI ĐÃ SỬ DỤNG TRONG MACRO

MACRO LỒNG NHAU

MỘT CÁCH ĐƠN GIẢN ĐỂ XÂY DỰNG MACRO LÀ XÂY DỰNG 1 MACRO MỚI TỪ MACRO ĐÃ CÓ.

EX : HIỂN THỊ 1 CHUỖI TẠI 1 TOẠ ĐỘ CHO TRƯỚC

```
DISPLAY_AT  MACRO ROW, COL, STRING
   LOCATE ROW, COL ;Gọi macro định vị  cursor
   DISPLAY STRING ; Gọi Macro xuất string
ENDM
```

MỘT MACRO CÓ THỂ THAM CHIẾU ĐẾN CHÍNH NÓ, NHỮNG MACRO NHƯ VẬY GỌI LÀ MACRO ĐỆ QUI.

ĐỊNH NGHĨA NHÃN BÊN TRONG MACRO

RONG MACRO CÓ THỂ CÓ NHÃN.

GỌI MACRO NHIỀU LẦN ➔ NHIỀU NHÃN ĐƯỢC TẠO RA

➔ LÀM SAO GIẢI QUYẾT VẤN ĐỀ NHẢY ĐIỀU KHIỂN?

SEMBLY GIẢI QUYẾT VẤN ĐỀ NÀY BẰNG CHỈ THỊ LOCAL
ỖNG BỨC MASM TẠO RA 1 TÊN DUY NHẤT CHO MỖI MỘT
BEL KHI MACRO ĐƯỢC GOI..

CÚ PHÁP : LOCAL LABEL_NAME

Một số Macro yêu cầu user định nghĩa các thành phần dữ liệu và các nhãn bên trong định nghĩa của Macro.

Nếu sử dụng Macro này nhiều hơn 1 lần trong cùng một chương trình, trình ASM định nghĩa thành phần dữ liệu hoặc nhãn cho mỗi lần sử dụng → các tên giống nhau lặp lại khiến cho ASM báo lỗi.

Để đảm bảo tên nhãn chỉ được tạo ra 1 lần, ta dùng chỉ thị LOCAL ngay sau phát biểu Macro

- Khi ASM thấy 1 biến được định nghĩa là LOCAL nó sẽ thay thế biến này bằng 1 ký hiệu có dạng ??n, trong đó n là 1 số có 4 chữ số. Nếu có nhiều nhãn có thể là ??0000, ??0001, ??0002 ...

Ta cần biết điều này để trong CT chính ta không sử dụng các biến hay nhãn đưới cùng 1 dạng.

Thí dụ minh họa chỉ thị Local

Xây dựng Macro REPEAT có nhiệm vụ xuất count lần số ký tự char ra màn hình.

```
REPEAT   MACRO  CHAR,  COUNT
 LOCAL  L1
MOV  CX,  COUNT
L1:  MOV  AH,2
        MOV  DL,  CHAR
        INT  21H
        LOOP  L1
ENDM
```

GIẢ SỬ GỌI :

REPEAT ‘A’, 10

REPEAT ‘*’, 20

ASM SẼ DÙNG CƠ CHẾ ĐÁNH SỐ CÁC NHÃN (TỪ 0000H ĐẾN FFFFH) ĐỂ ĐÁNH DẤU CÁC NHÃN CÓ CHỈ ĐỊNH LOCAL.

SẼ ĐƯỢC DỊCH RA ➔

Thí dụ minh họa chỉ thị Local

```
MOV CX, 10
??0000 : MOV AH,2
 MOV DL, 'A'
 INT 21H
LOOP ??0000
MOV CX, 20
??0001 : MOV AH,2
MOV DL, '*'
INT 21H
LOOP ??0001
```

Thí dụ minh họa

Viết 1 macro đưa từ lớn hơn trong 2 từ vào AX

```
GETMAX    MACRO    WORD1,  WORD2
      LOCAL   EXIT
    MOV  AX,  WORD1
   CMP  AX,  WORD2
  JG  EXIT
  MOV  AX,  WORD2
  EXIT  :
ENDM
```

GIẢ SỬ FIRST,SECOND, THIRD LÀ CÁC BIẾN WORD.

SỰ THAM CHIẾU MACRO ĐƯỢC MỞ RỘNG NHƯ SAU :

```
MOV AX, FIRST
CMP AX, SECOND
JG ??0000
MOV AX, SECOND
??0000:
```

Thí dụ minh họa

Viết 1 macro đưa từ lớn hơn trong 2 vào AX

LỜI GỌI MACRO TIẾP THEO :

GETMAX SECOND, THIRD

ĐƯỢC MỞ RỘNG NHƯ SAU :

MOV AX, SECOND

CMP AX, THIRD

JG ??0001

??0001 :

SỰ THAM CHIẾU LIÊN TIẾP MACRO NÀY HAY ĐẾN MACRO KHÁC KHIẾN TRÌNH BIÊN DỊCH CHÈN CÁC NHÃN ??0002, ??0003 VÀ CỨ NHƯ VẬY TRONG CHƯƠNG TRÌNH CÁC NHÃN NÀY LÀ DUY NHẤT.

THƯ VIỆN MACRO

CÁC MACRO MÀ CHƯƠNG TRÌNH THAM CHIẾU CÓ THỂ ĐẶT Ở FILE RIÊNG ➔ TA CÓ THỂ TẠO 1 FILE THƯ VIỆN CÁC MACRO.

- **DÙNG 1 EDITOR ĐỂ SOẠN THẢO MACRO**
- **LƯU TRỮ TÊN FILE MACRO.LIB**
- **KHI CẦN THAM CHIẾU ĐẾN MACRO TA DÙNG CHỈ THị INCLUDE TÊN FILE THƯ VIỆN**

MỘT CÔNG DỤNG QUAN TRỌNG CỦA MACRO LÀ TẠO RA CÁC LỆNH MỚI.

SO SÁNH GIỮA MACRO & THỦ TỤC

- THỜI GIAN BIÊN DỊCH.

MACRO ÍT TỐN THỜI GIAN BIÊN DỊCH HƠN PROCEDURE

- THỜI GIAN THỰC HIỆN : NHANH HƠN PROCEDURE VÌ KHÔNG TỐN THỜI GIAN KHÔI PHỤC TRẠNG THÁI THÔNG TIN KHI ĐƯỢC GỌI → TỐC ĐỘ NHANH HƠN.

- KÍCH THƯỚC : KÍCH THƯỚC CT DÀI HƠN

CÁC LỆNH LẶP TRONG MACRO

- **REP <BIỂU THỨC> :**

  ...

  ENDM

- **TÁC DỤNG : LẶP LẠI CÁC KHỐI LỆNH TRONG MACRO VỚI SỐ LẦN LÀ <BIỂU THỨC>**

```
EX : MSHL MACRO OPER, BITS
        REPT BITS
        SHL DEST, 1
     ENDM
ENDM
```

GỌI MSHL BX, 3

SẼ ĐƯỢC THAY THẾ BẰNG :

SHL BX, 1

SHL BX, 1

SHL BX, 1

CÁC LỆNH LẶP TRONG MACRO

- IRP <THÔNG SỐ>, <DANH SÁCH CÁC TRỊ TRONG NGOẶC NHỌN> :

  ...

  ENDM

TÁC DỤNG :

- LẶP LẠI KHỐI LỆNH TÙY THEO DANH SÁCH TRỊ.
- SỐ LẦN LẶP CHÍNH LÀ SỐ TRỊ TRONG DANH SÁCH
- MỖI LẦN LẶP LẠI SẼ THAY <THÔNG SỐ> BẰNG 1 TRỊ TRONG DANH SÁCH VÀ SẼ LẦN LƯỢT LẤY HẾT CÁC TRỊ TRONG DANH SÁCH.

```
EX : PROCTABLE   LABEL   WORD
       IRP PROCNAME, <MOVEUP, MOVDOWN,MOVLEFT,MOVRGHT>
       DW   PROCNAME
     ENDM
```

CÁC LỆNH LẶP TRONG MACRO

- TUY NHIÊN CÁCH KHAI BÁO NÀY RƯỜM RÀ HƠN LÀ DÙNG :

  PROCTABLE DW MOVUP, MOVDOWN,MOVLEFT,MOVRIGHT

➔ VIỆC SỬ DỤNG CÁC MACRO LẶP VÒNG NÀY CHO CÓ HIỆU QUẢ LÀ ĐIỀU KHÓ, ĐÒI HỎI PHẢI CÓ NHIỀU KINH NGHIỆM

BÀI TẬP MACRO

Bài 1 : 1. Viết một MACRO tính USCLN của 2 biến số M và N. Thuật toán USCLN như sau :

```
WHILE    N <> 0 DO
         M = M MOD N
         Hoán vị  M và N
END_WHILE
```

Bài 2 : MACRO doi tu so chua trong ax sang chuoi tro den boi DI

```
;     in :   DI =offset   chuoi
;              AX =so can doi
;     out: khong co(chuoi van do di tro toi)
```

Bài 3 :Viết macro chuyen tu chuoi thanh so chua trong ax

; in : DI =offset chuoi

; out : AX =so da doi

Bài 4 : Viết MACRO xuất số hecxa chứa trong AL ra màn hình *

; INPUT : AL chứa số cần xuất; OUTPUT: nothing

Bài 5 : Viết Macro in số hecxa chứa trong BL ra dạng binary

;Input: BL chứa số cần in

;Output: Nothing