

บทที่ 17 คำสั่งตารางและการสร้างแมโคร

- คำสั่ง xlat
- การสร้าง Macro
 - Macro กับ Parameter
 - Macro local variable
 - Macro กับโปรแกรมย่อย

คำสั่งเปิดตาราง

- คำสั่ง **XLAT**

- คำสั่งเปิดตารางนี้ทำงานเหมือนคำสั่ง

```
mov    al, [bx+al]
```

EX เขียนโปรแกรมแปลงเลขค่าใน AL เป็นอักขระฐานสิบ
หกหนึ่งหลัก

```
.data
```

```
hextable db '0123456789ABCDEF'
```

```
...
```

```
.code
```

```
...
```

```
mov    bx, offset hextable  
xlat
```

ตัวอย่าง

- จงเขียนโปรแกรมย่อยที่รับค่าเป็นตัวเลขโคคฐานสิบหนึ่งตัวทางรีจิสเตอร์ AL แล้วแปลงเลขดังกล่าวเป็นเลขที่มีลำดับตรงกันข้ามกันแล้วให้ผลลัพธ์ในรีจิสเตอร์ AL (เช่น 0 จะแปลงเป็น 9, 1 จะแปลงเป็น 8 เป็นต้น)

```
.data
seq      db      9,8,7,6,5,4,3,2,1,0
.code
conv     proc     near
          push     bx
          mov      bx, offset seq
          xlat
          pop      bx
          ret
        endp

...

```

แมคโคร

- แมคโคร
 - ส่วนของโปรแกรมที่ได้รับการตั้งชื่อไว้.
 - เมื่อเราเรียกใช้ชื่อนั้นในโปรแกรมส่วนของโปรแกรมที่มีชื่อดังกล่าวจะถูกนำไปแทนที่.

- รูปแบบการประกาศ

```
name          MACRO  parameters  
                ...  
                ENDM
```

```


|    |
|----|
| EX |
|----|

print macro  
    mov      bx,offset msg  
    mov      ah,09h  
    int      21h  
    endm  
  
    .data  
msg      db      'TEST$'  
    .code  
  
    ...  
    print
```

แมคโคร

- เมื่อโปรแกรมถูกแปลโปรแกรมที่ได้จะมีลักษณะดังนี้.

```
.data
msg      db      'TEST$'
.code
;          ...
          print
mov       bx,offset msg
mov       ah,09h
int       21h
          ...
```

พารามิเตอร์ในแมคโคร

- พารามิเตอร์มีลักษณะคล้ายกับแมคโครซ้อนอยู่ในแมคโครอีกทีหนึ่ง
- พารามิเตอร์แต่ละตัวจะถูกนำไปแทนที่ในทุกจุดในแมคโครที่เรียกใช้มัน

EX

```
print macro prt
mov     bx,offset prt
mov     ah,09h
int     21h
endm

.data
msg1     db     'TEST$'
msg2     db     'TEST2$'
.code

...
print    msg1
print    msg2
```

พารามิเตอร์ในแมคโคร

- เมื่อโปรแกรมถูกแปลโปรแกรมที่ได้จะมีลักษณะดังนี้.

```
.data
msg1      db      'TEST$'
msg2      db      'TEST2$'
.code

;          ...
;          print msg1
;          mov     bx,offset msg1
;          mov     ah,09h
;          int     21h

;          print msg2
;          mov     bx,offset msg2
;          mov     ah,09h
;          int     21h
;          ...
```

แมคโครที่ใช้ LABEL

- ถ้าภายในแมคโครมีการใช้ LABEL

```
test1 macro
    mov     ...
label1:
    ...
endm

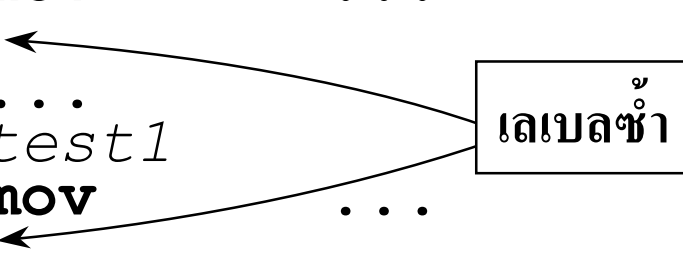
.code

    test1
    test1
    ...
```

- เมื่อแมคโครได้รับการขยายจะมีลักษณะดังนี้

```
.code

;    test1
    mov     ...
label1: ←
;    test1
    mov     ...
label1: ←
    ...
```



เลเบลซ้ำ

แมคโครที่ใช้ LABEL

- เราสามารถกำหนดให้ MASM สร้าง LABEL ที่มีชื่อไม่ซ้ำกันจากเลเบลของเราได้
- เราจะต้องประกาศให้เลเบลเป็นแบบ LOCAL

```
test1 macro
    local    label1
    mov      ...
label1:
    ...
endm
```

ลักษณะของโปรแกรมที่ขยายแล้วจะเป็นดังนี้

```
.code

;          ...
          test1
xxxx1:   mov      ...

;          ...
          test1
xxxx2:   mov      ...

          ...
```

แมคโครกับโปรแกรมย่อย

- ข้อแตกต่าง
 - คำสั่งในแมคโครจะถูกนำไปแทนที่ในตำแหน่งที่มีการเรียกใช้. ในการทำงานจริงจะไม่มีกรกระโดดไปทำงาน แต่ถ้ามีการเรียกใช้แมคโครหลายครั้ง โปรแกรมส่วนนั้นก็จะถูกขยายออกมาหลายครั้ง.
 - โปรแกรมจะเป็นส่วนของโปรแกรมที่มีชุดเดียว. การเรียกใช้โปรแกรมย่อยทุกครั้งจะเป็นการกระโดดไปทำงานในโปรแกรมย่อยนั้น.
- ข้อดีของแมคโคร
 - ลดเวลาในการกระโดดไปทำงานสำหรับส่วนของคำสั่งที่มีขนาดเล็ก
- ข้อเสียของแมคโคร
 - ถ้าส่วนของคำสั่งมีขนาดใหญ่ หรือมีการใช้หลายครั้ง จะทำให้โปรแกรมที่ได้จากการแปลมีขนาดใหญ่ด้วย.