

บทที่ 8 : การประกาศข้อมูล

- รูปแบบในการประกาศข้อมูล
- การอ้างใช้ข้อมูล
- การประกาศข้อมูลสำหรับการเรียกใช้งาน
บริการของ DOS หมายเลข 09h และ 0Ah

การประกาศข้อมูลหรือตัวแปร

- ข้อมูลที่โปรแกรมใช้ และ ตัวแปรต่าง ๆ ที่อยู่ในหน่วยความจำ ➡ เป็นสิ่งเดียวกัน.
- การประกาศข้อมูล =
 - การระบุให้ assembler จองเนื้อที่ในหน่วยความจำไว้สำหรับการเก็บข้อมูล
 - และตั้งชื่อให้กับหน่วยความจำตำแหน่งนั้น (สร้างเลเบล)
- ตัวอย่างการประกาศตัวแปรแบบเลขจำนวนเต็ม 16 บิต 2 ตัว (i, j) และอักขระ 1 ไบต์ (ch)

i :		จะเป็นการจองเนื้อที่ ในหน่วยความจำ ขนาด 5 ไบต์ โดยจะมี เลเบล i j และ ch ชี้ ตำแหน่งดังรูป
j :		
ch :		

คำสั่งเทียมสำหรับการจองเนื้อที่ในหน่วยความจำ

- เราใช้คำสั่งเทียมในการระบุกับ assembler ว่าต้องการหน่วยความจำขนาดหน่วยละเท่าใด.
- คำสั่งต่าง ๆ

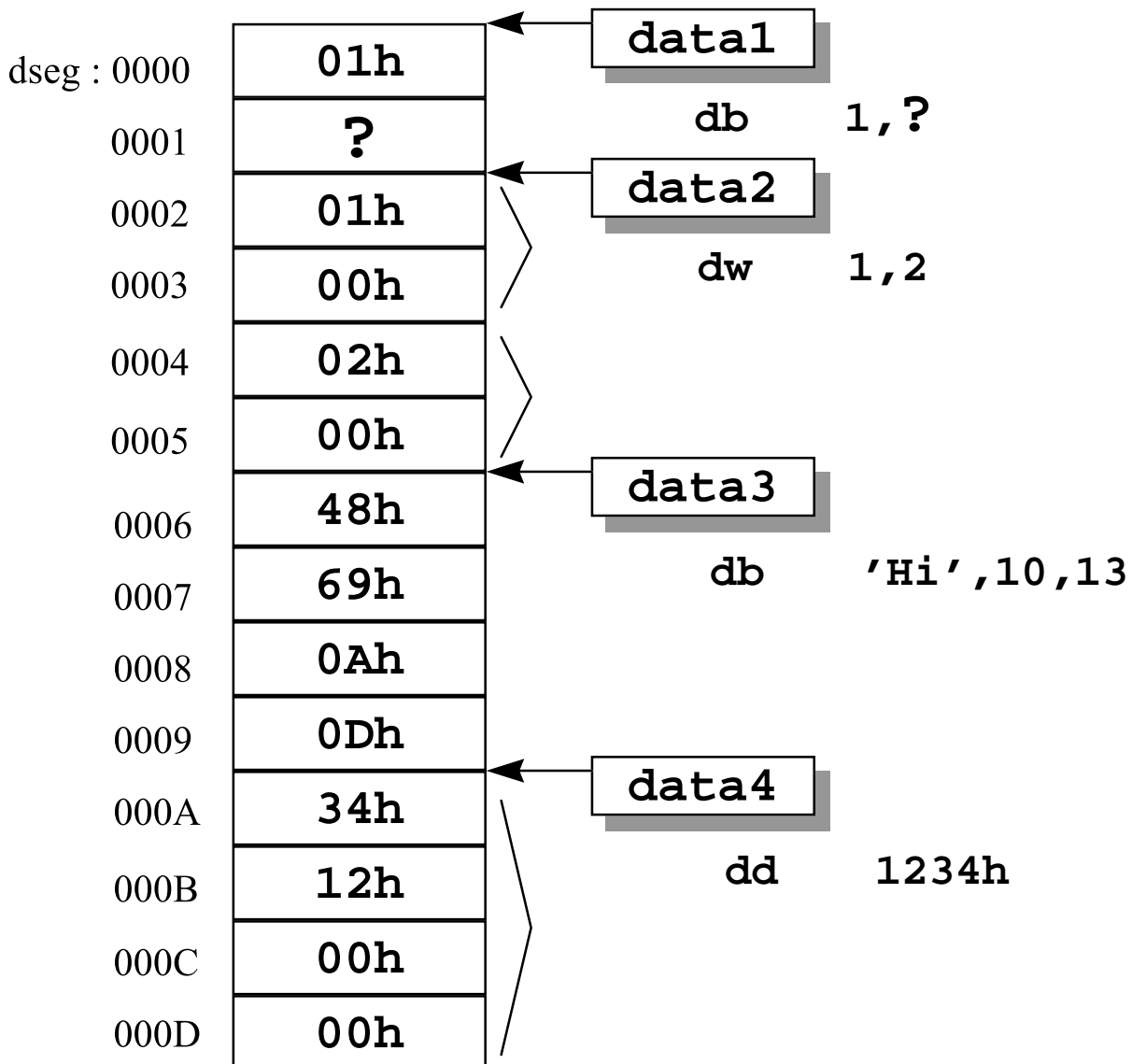
DB	Define Byte
DW	Define Word
DD	Define Doubleword
DQ	Define Quadword
DT	Define Tenbytes

ระบุขนาดของหน่วยย่อยของการจองหน่วยความจำ

- ตัวอย่างของการจองหน่วยความจำ

dseg	segment
data1	db 1, ?
data2	dw 1, 2
data3	db 'Hi', 10, 13
data4	dd 1234h
dseg	ends

การจัดเรียงค่าในหน่วยความจำ



?

สังเกตว่าในการประกาศ data1 ไบต์ที่สอง เราใช้เครื่องหมาย '?' ในการระบุการจองเนื้อที่หน่วยความจำโดยไม่ระบุค่าของข้อมูล.

การระบุค่าของหน่วยความจำ ที่มีการซ้ำ

- เราใช้คำสั่งเทียบ dup ในการระบุค่าเริ่มต้นที่มีการซ้ำกันหลายชุดค่า.

- รูปแบบ

จำนวนซ้ำ **dup** (ค่าที่ซ้ำ)

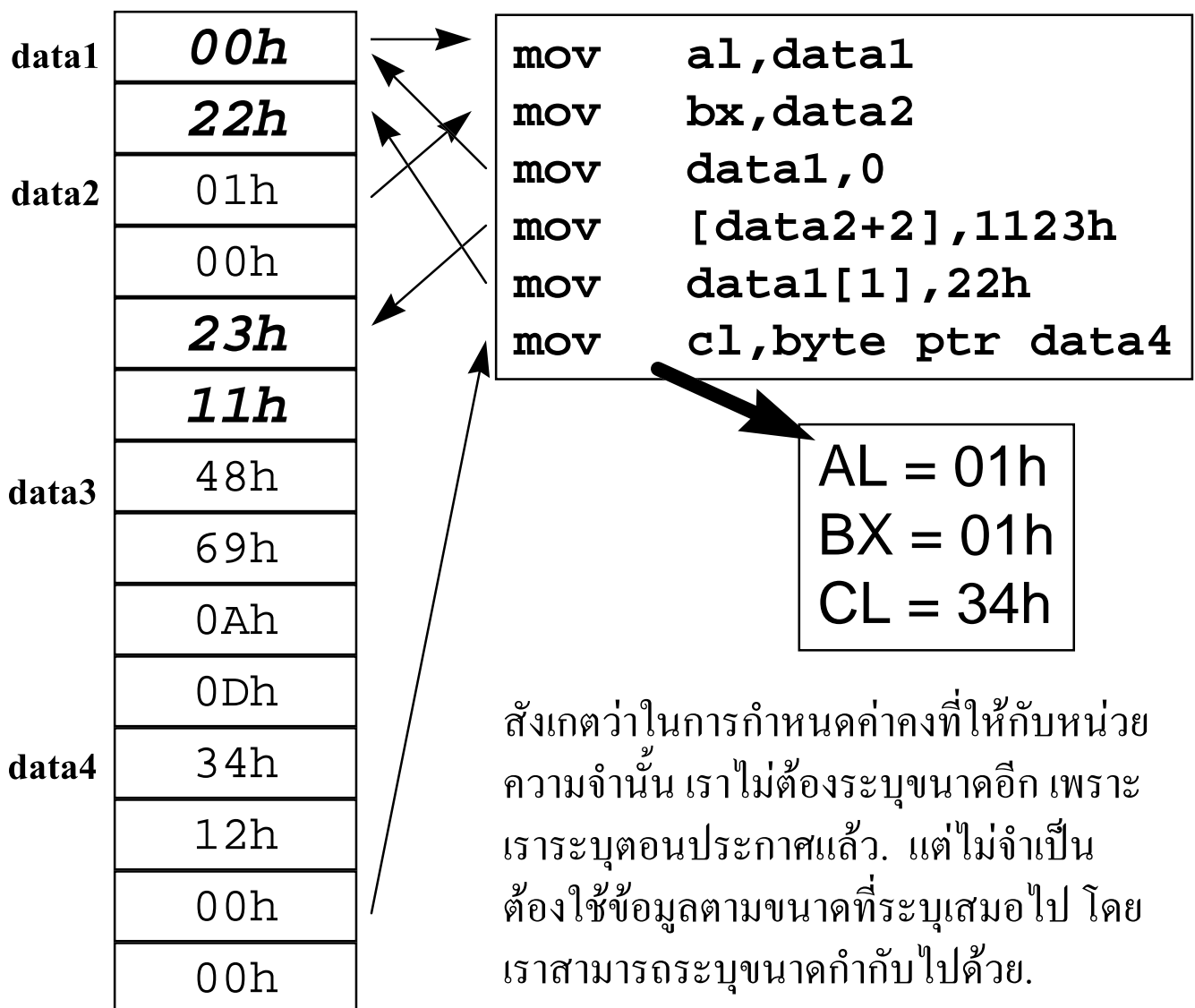
- ตัวอย่าง

```
data7      db    10 dup (0)
data8      db    5 dup (4 dup (5))
data9      db    4 dup (1, 2, 3 dup (4))
data10     db    20 dup (?)
```

data7	0	0	0	0	0	-----ซ้ำ----->	0			
						-----ซ้ำ----->				
data8	5	5	5	5	5	-----ซ้ำ----->				
						-----ซ้ำ----->	5			
data9	1	2	4	4	4	1	2	4	4	4
	1	2	4	4	4	1	2	4	4	4
data10	?	?	?	?	?	-----ซ้ำ----->				
						-----ซ้ำ----->	?			

การอ้างอิงข้อมูลในหน่วยความจำ

- เราสามารถใช่เลเบลที่ประกาศแทนตำแหน่งของ
มูลในหน่วยความจำได้ทันที. เรายังสามารถอ้าง
หน่วยความจำโดยคิดสัมพัทธ์กับตำแหน่งของ
เลเบลได้.



การอ้างอิงข้อมูลในหน่วยความจำ

- การอ้างตำแหน่งของข้อมูล
 - ใช้คำสั่งเทียม **OFFSET**

```
mov ax,data2           ;ค่าในหน่วยความจำ
mov bx,offset data2    ;ออฟเซตของข้อมูล
```

- การใช้รีจิสเตอร์ BX ในการอ้างตำแหน่งข้อมูล

```
mov bx,offset data7    ;ออฟเซตของข้อมูล
mov byte ptr [bx],10
mov byte ptr [bx+1],20
```

```
mov bx,2
mov data7[bx],30
mov data7[bx+1],40
```

- ในโปรแกรมช่วงแรก BX เก็บออฟเซตของ DATA7 ไว้ จากนั้นเราอ้างอิงหน่วยความจำเทียบกับ BX.
- ในส่วนหลังเราอ้างอิงหน่วยความจำเทียบกับ DATA7 และค่าใน BX.

ตัวอย่าง

จากการประกาศข้อมูลต่อไปนี้

```
dseg    segment
data1    db      1, 2, 3
data2    db      3 dup ( ? )
data3    dw      5, 6
dseg    ends
```

01	DS:0000
02	DS:0001
03	DS:0003
?	DS:0004
?	DS:0005
?	DS:0006
05	DS:0007
00	DS:0008
06	DS:0009
00	DS:000A

```
mov     al, data1
```

- AL = 01h

```
mov     bx, offset data1
```

- BX = 0000h

```
mov     cx, data3[2]
```

- CX = 0006h

```
mov     bx, offset data2
```

```
mov     [bx], cx
```

- [DS:0004] = 06h

- [DS:0005] = 00h

การประกาศข้อมูลสำหรับการใช้ บริการหมายเลข 09h ของ DOS

บริการหมายเลข 09h : แสดงผลข้อความ

- รับข้อมูลผ่านรีจิสเตอร์ดังนี้
 - AH = 09h
 - DS : DX = ตำแหน่งของข้อความที่จะแสดง
โดยข้อความนี้ต้องจบด้วยเครื่องหมาย '\$'

- เราสามารถประกาศข้อมูลได้ดังนี้

```
dseg    segment
msg      db    'hello',10,13,'$'
dseg    ends
```

โดยรหัส 10 (Line Feed) และ 13 (Carriage Return)
เป็นรหัสที่สั่งให้ขึ้นบรรทัดใหม่.

- การสั่งพิมพ์

```
mov      ah,9
mov      dx,offset msg
int      21h
```

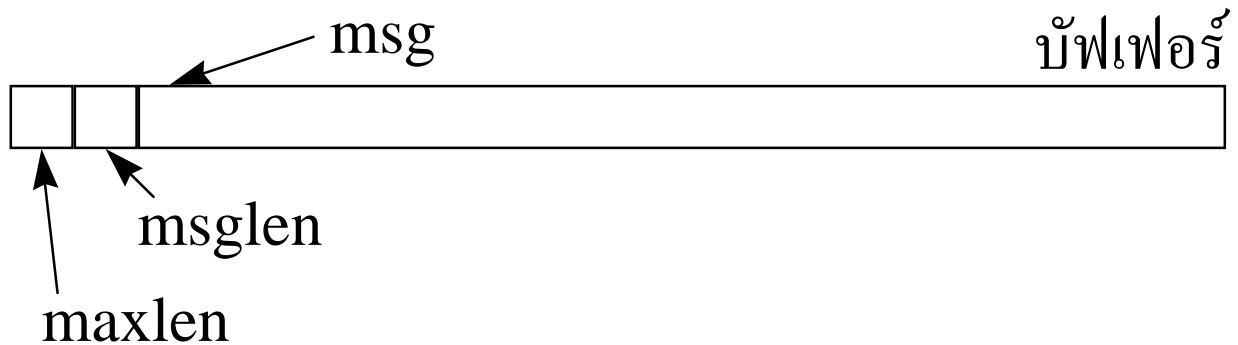
การประกาศข้อมูลสำหรับการใช้ บริการหมายเลข 0Ah ของ DOS

บริการหมายเลข 0Ah : อ่านข้อความ

- รับข้อมูลผ่านรีจิสเตอร์ดังนี้
 - AH = 09h
 - DS : DX = ตำแหน่งของหน่วยความจำที่จะใช้เก็บข้อมูล (บัฟเฟอร์) โดยที่จะต้องมีรูปแบบดังนี้
 - ไบต์ที่ 1 : ความยาวของข้อมูลทั้งหมด (ต้องเผื่อที่ว่างไว้ 1 ไบต์สำหรับอักขระจิ้นบรรทัดใหม่)
 - ไบต์ที่ 2 : DOS จะใส่ความยาวข้อมูลจริงลงที่หน่วยความจำตำแหน่งนี้
 - ไบต์ต่อ ๆ ไป : ข้อความที่อ่านได้. เราต้องจองเนื้อที่หน่วยความจำให้เพียงพอ
- ตัวอย่างของการประกาศข้อมูลเพื่อให้เขียนโปรแกรมได้สะดวก

```
.data
maxlen      db      30
msglen      db      ?
msg         db      30 dup ( ? )
```

การประกาศข้อมูลสำหรับการใช้ บริการหมายเลข 0Ah ของ DOS



ในการส่งแอดเดรสของบัฟเฟอร์ไปยัง DOS เราจะส่งตำแหน่งของ maxlen ไปให้ เพื่อให้รูปแบบของข้อมูลตรงตามข้อกำหนด.

- การเรียกใช้บริการอ่านข้อความ

```
mov    ah, 0Ah
mov    dx, offset maxlen
int    21h
```

- msglen จะเก็บความยาวของข้อความที่อ่านได้.
- อักขระต่างในข้อความจะถูกเก็บในหน่วยความจำตั้งแต่ msg เป็นต้นไป.

ตัวอย่างโปรแกรม

- โปรแกรมที่รับข้อความจากผู้ใช้แล้วแสดงข้อความนั้นออกมา.
- รับข้อความ
 - บริการหมายเลข 0Ah
- แสดงข้อความ
 - บริการหมายเลข 09h
 - ข้อความต้องจบด้วย '\$'