

แผนบริหารการสอนประจำบทที่ 10

หัวข้อเนื้อหา

- การสร้างโครงสร้างการตัดสินใจแบบ if-then-else
- การสร้าง repeat until loop
- การสร้าง while loop
- การสร้าง for loop

วัตถุประสงค์เชิงพฤติกรรม

- มีความรู้และความเข้าใจเกี่ยวกับรูปแบบของโปรแกรมภาษาแอสเซมบลีที่มีโครงสร้างแบบต่าง ๆ เช่น if-then-else, repeat until loop, while loop และ for loop เป็นต้น

วิธีสอนและกิจกรรมการเรียนการสอน

- บรรยาย
- สืบเสาะหาความรู้
- ค้นคว้าเพิ่มเติม
- ตอบคำถาม

สื่อการเรียนการสอน

- สื่ออิเล็กทรอนิกส์
- ตอบคำถาม
- ภาพ
- เอกสารอ้างอิงประกอบการค้นคว้า

การวัดผลและประเมินผล

ใช้วิธีการสังเกตและจดบันทึกไว้เป็นระยะ

- สังเกตจากงานที่กำหนดให้ไปทำมาส่ง
- สังเกตจากการตอบคำถาม
- สังเกตจากการนำความรู้ไปใช้

การประเมินผล

วิธีตรวจผลงานต่างๆ ที่ให้ทำ

- ตรวจผลงานภาคปฏิบัติ
- ตรวจรายงาน
- ตรวจแบบฝึกหัด

ใช้วิธีการออกข้อสอบข้อเขียน

บทที่ 10 โครงสร้างควบคุม (Control Structure)

เราได้ศึกษาคำสั่งกระโดดในบทที่ 9 โดยคำสั่งกลุ่มนี้ทำให้เราสามารถเขียนโปรแกรมให้มีการทำงานที่ซับซ้อนขึ้นได้ แต่การใช้คำสั่งเหล่านี้อย่างไม่เป็นระบบทำให้โปรแกรมที่เขียนขึ้นนั้นทำความเข้าใจได้ยาก และมีลักษณะเหมือนเส้นสปาเก็ตตี้ได้ ในบทนี้เราจะพิจารณาการใช้คำสั่งกระโดดมาสร้างเป็นโครงสร้างควบคุมรูปแบบต่าง ๆ การใช้คำสั่งกระโดดในลักษณะนี้จะทำให้โปรแกรมของเรามีความเป็นโครงสร้างมากขึ้น

การสร้างโครงสร้างการตัดสินใจแบบ if-then-else

รูปแบบของโครงสร้างที่ง่ายที่สุดคือโครงสร้างแบบ if-then-else รูปแบบของโปรแกรมภาษาแอสเซมบลีให้มีโครงสร้างแบบ if-then-else มีลักษณะดังนี้

```

    if condition is false then jump to elselabel
        then_actions
        jump to endif_label
    elselabel:
        else_actions
    endif_label:
  
```

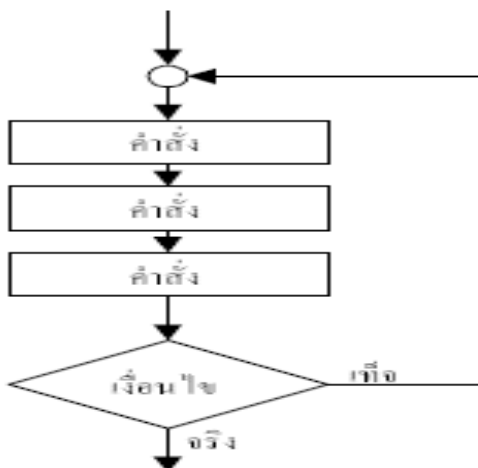
ตัวอย่างของโปรแกรมมีลักษณะโปรแกรมที่ 10.1

(a)	(b)
<pre> if AL<10 then DL:=AL+'0' else DL:=AL+'A'-10; </pre>	<pre> cmp al,10 jae abovenine mov dl,al add dl,'0' jmp endif abovenine: mov dl,al add al,'A'-10 endif: </pre>

โปรแกรมที่ 10.1 โปรแกรม (a) แสดงตัวอย่างของโปรแกรมที่เขียนด้วยโครงสร้างของภาษา pascal
โปรแกรม (b) แสดงโปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากัน

การสร้าง repeat until loop

โครงสร้างของ repeat until loop ในภาษาระดับสูงทั่วไปมีลักษณะดังรูปที่ 10.1



รูปที่ 10.1 โครงสร้างควบคุมแบบ repeat until

เราสามารถเขียนโปรแกรมโดยใช้ภาษาแอสเซมบลีโดยมีโครงสร้างแบบ repeat until ได้ ดังตัวอย่าง

(a)

```

BL:=1;
CX:=0;
DX:=0;
Repeat
DX:=DX+BL*BL;

BL:=BL+1;
CX:=CX+1;
until (DX>100);
  
```

(b)

```

mov bl,1
mov cx,0
mov dx,0
startloop:
mov al,bl
mul bl ;result in ax
add dx,ax
inc bx
inc cx
cmp dx,100
jbe startloop
  
```

โปรแกรมที่ 10.2 โปรแกรม (a) แสดงตัวอย่างของโปรแกรมภาษา Pascal ที่ใช้โครงสร้างแบบ repeat until
 โปรแกรม (b) แสดงโปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากัน

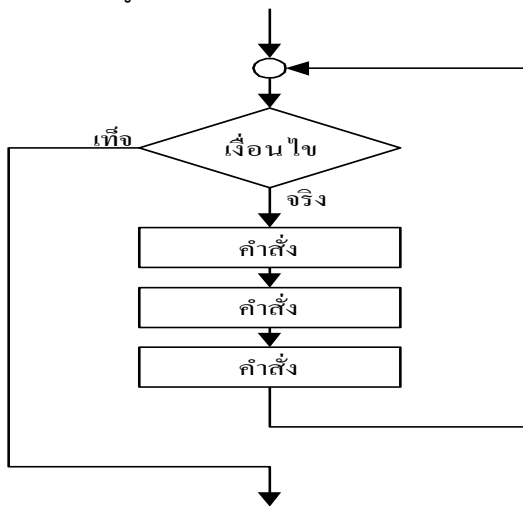
รูปแบบของโปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากับ repeat until loop มีลักษณะเป็นดังนี้

```

startlabel:
  action;
  ...
  action;
  if condition is false then jump to startlabel
  
```

การสร้าง while loop

โครงสร้างของ while loop ในภาษาระดับสูงทั่วไปมีลักษณะเป็นดังนี้



รูปที่ 10.2 โครงสร้างควบคุมแบบ while

เราสามารถเขียนโปรแกรมโดยใช้ภาษาแอสเซมบลีโดยมีโครงสร้างแบบ while loop ได้ ดังตัวอย่าง

(a)

```

while (DL<>13) and
      (CX<20) do
begin
  AX:=AX+DL;

  BX:=BX+1;
  DL:=DATA[BX];
  CX:=CX+1;

end;
    
```

(b)

```

startloop:
  cmp dl,13
  jz  endloop
  cmp cx,20
  jae endloop

  add al,dl
  adc ah,0
  inc bx
  mov dl,data[bx]
  inc cx
  jmp startloop
endloop:
...
action
jump to startlabel
endlabel:
    
```

โปรแกรมที่ 10.3 โปรแกรม (a)

แสดงตัวอย่างของโปรแกรม ภาษา pascal ที่ใช้โครงสร้างแบบ while loop

โปรแกรม (b) แสดงโปรแกรม ภาษาแอสเซมบลีที่เทียบเท่ากัน

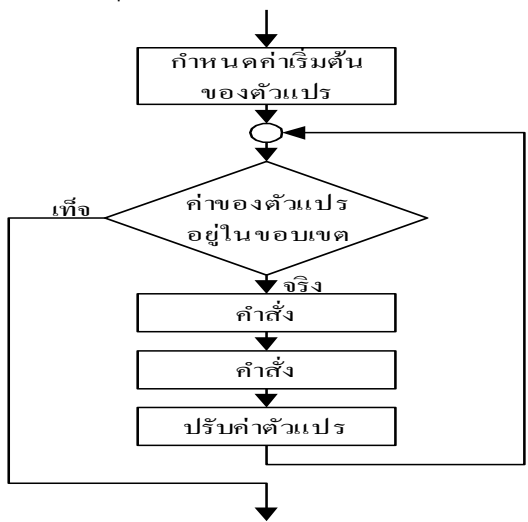
รูปแบบของโปรแกรม ภาษาแอสเซมบลีที่เทียบเท่ากับ while loop มีลักษณะเป็นดังนี้

```

startlabel:
if condition is false then
  jump to endlabel
action
    
```

การสร้าง for loop

เราสามารถใส่คำสั่งกระโดดในการสร้างโครงสร้างแบบ for loop ได้ นอกจากนั้นเรายังสามารถใช้คำสั่งกลุ่ม LOOP ในการสร้างโครงสร้างแบบ for loop ได้เช่นเดียวกัน โครงสร้างของ for loop มีลักษณะดังรูปที่ 10.3



รูปที่ 10.3 โครงสร้างของ for loop

รูปแบบของโปรแกรมภาษาแอสเซมบลีที่มีโครงสร้างเป็นแบบ for loop มีลักษณะดังนี้

initialize index variables

startloop:

*if index value is not in the range then jump to endloop
action*

...

action

update index variable

jump to startloop

endloop:

ตัวอย่างของโปรแกรมที่เขียนโดยใช้รูปแบบลักษณะนี้เป็นดังโปรแกรมที่ 10.4.

(a)

CX:=0;

for DL:=1 to 100 do

(b)

mov cx,0

mov dl,1

startloop:

```

                                cmp dl,100
                                ja  endloop

begin
if DL mod 7=0 then                mov al,dl
                                mov ah,0
                                mov bl,7
                                div bl

                                cmp ah,0
                                jne endif

                                CX:=CX+1;          inc cx
                                endif:

                                End;                inc dl
                                                jmp startloop
                                                endloop:

```

โปรแกรมที่ 10.4 โปรแกรม (a) แสดงตัวอย่างของโปรแกรมภาษา pascal ที่ใช้โครงสร้างแบบ for loop
โปรแกรม (b) แสดงโปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากัน

เรายังสามารถใช้คำสั่ง LOOP ในการสร้างโครงสร้างแบบ for loop ได้ เช่นเดียวกัน ดังต่อไปนี้

```

                                set the value of CX
startloop:
                                actions
                                LOOP startloop

```

แต่การใช้คำสั่ง LOOP ในการสร้างโครงสร้างแบบ for loop ไม่สามารถสร้างโครงสร้างการกระทำซ้ำที่มีความซับซ้อนมาก ๆ ได้ เช่นการกระทำซ้ำที่มีวงรอบของการกระทำซ้ำซ้อนกันหลาย ๆ วง

สำหรับคำสั่ง LOOPZ และ LOOPNZ นั้น เราสามารถนำมาใช้ในการสร้างโครงสร้างควบคุมที่มีความซับซ้อนขึ้นได้ โดยโครงสร้างดังกล่าวจะมีลักษณะปนกันระหว่าง for loop และ while loop หรือ repeat until นั่นคือเงื่อนไขควบคุมการกระทำซ้ำจะขึ้นกับทั้งค่าของรีจิสเตอร์ (มีลักษณะคล้ายโครงสร้างแบบ for loop) และเป็นเงื่อนไขจริง ๆ (คล้าย repeat until loop และ while loop) ดังในตัวอย่างโปรแกรมต่อไปนี้

(a)	(b)
AX:=0;	mov ax,0
CX:=100;	mov cx,100
repeat	startloop:
AX:=AX+data[BX];	add ax,data[BX]
BX:=BX+2;	add bx,2
CX:=CX-1;	cmp data[BX],0
until (data[BX]=0) or (CX=0);	loopnz startloop

โปรแกรมที่ 10.5 โปรแกรม (a) โปรแกรมที่เขียนด้วยภาษา pascal
โปรแกรม (b) โปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากัน

สรุป

การใช้คำสั่งกระโดดมาสร้างเป็นโครงสร้างควบคุมรูปแบบต่าง ๆ การใช้คำสั่งกระโดดในลักษณะนี้จะทำให้โปรแกรมของเรามีความเป็นโครงสร้างมากขึ้น เช่น การสร้างโครงสร้างการตัดสินใจแบบ if-then-else, repeat until loop, while loop และ for loop เป็นต้น ดังนั้นผู้เขียนโปรแกรมภาษาแอสเซมบลีจำเป็นต้องทำความเข้าใจวิธีการและรูปแบบการเลือกการสร้างโครงสร้างควบคุมใช้งานให้เหมาะสมมากที่สุด

คำถามทบทวน

1. อธิบายพร้อมยกตัวอย่างการทำงานของโครงสร้างควบคุมรูปแบบ if-then-else
2. อธิบายพร้อมยกตัวอย่างการทำงานของโครงสร้างควบคุมรูปแบบ repeat until loop
3. อธิบายพร้อมยกตัวอย่างการทำงานของโครงสร้างควบคุมรูปแบบ while loop
4. อธิบายพร้อมยกตัวอย่างการทำงานของโครงสร้างควบคุมรูปแบบ for loop
5. จงเปรียบเทียบข้อดีและข้อเสียของรูปแบบโครงสร้างควบคุมแบบ repeat until และ while loop
6. จงเปลี่ยนโปรแกรมที่เขียนด้วยภาษาโปรแกรม pascal ที่ให้มาให้อยู่ในรูปของโปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากัน

```

AX:=20;
CX:=100;
repeat
  AX:=AX+data[BX];
  BX:=BX+8;
  CX:=CX-2;
until (data[BX]=0) or
  (CX=0);

```