الموجهات

تهدف الموجهات بصورة رئيسية إلى تحديد مجالات من الذاكرة لحجز مواقع بيانات بصورة جماعية, ويتم استرداد تلك البيانات كما في الموجهات في لغات البرمجة المختلفة. يتم التعامل مع المسجل SI بهدف تمرير البيانات من المواقع المختلفة من الذاكرة إلى السجلات التي نرغب بالتخزين ضمنها.

تمرين 1:

المطلوب كتابة برنامج يجمع الأعداد من 1إلى 10 وتخزين النتيجة في خلية يطلق عليها اسم SUM.

الحل:

يجب في البداية (كما هو مطلوب حجز مجال في الذاكرة نسميه sum) كما يلي:

```
 org 100h
      jmp start
sum  db 0
start:
      mov al, 0
ree:
      inc al
      add [sum], al
      cmp al,0ah
      jnz ree
```

ret

تمرين 2: يطلب كتابة برنامج يقوم بجمع مصفوفتين لهما القيم التالية:

vec1 db 1, 2, 5, 6

vec2 db 3, 5, 6, 1

الحل:

بما أنه لدينا مصفوفتين لذلك يجب حجز هذه المصفوفات على شكل موجهات ومن ثم حجز مجال مصفوفة خرج سيتم تخزين النتائج ضمنها, يتم تعريف مجال تخزين الخرج ويمكن أن نضع فيه قيمة ابتدائية صفرية كما في المثال السابق أو يمكن تحديد عبارة (بغض النظر عن القيمة) وبالتالي فإننا نقوم بوضع ?.

1. Define vec1, vec2, vec3
2. Load ea vec1.

1. sum← al + sum
2. Al=10?
3. No; Go To 2
4. End.

al; effective address

```
org 100h
    jmp start
    vec1 db 1, 2, 5, 6
    vec2 db 3, 5, 6, 1
    vec3 db ?, ?, ?, ?
    start:
    lea si, vec1
    lea bx, vec2
    lea di, vec3
    mov cx, 4
    sum:
      mov al, [si]
      add al, [bx]
      mov [di], al
      inc si
      inc bx
      inc di
      loop sum
```

تسمح التعليمة LEA بتحديد العناوين الفعالة التي تم تخزين الموجهين vec1,vec2 أي سيتم تخزين العنوان الذي تم حجزه للشعاع vec1 بالسجل SI وعنوان vec2 في السجل bx والموجه الخرج في di.

تم تحميل السجل cx بالقيمة 4 وذلك لأن عملية الجمع ستتكرر أربعة مرات (عدد عناصر المصفوفة) وسيتم استخدام الحلقة loop التي تتعامل مع cx وتقوم بإنقاص قيمتها في كل تكرار حتى الوصول إلى الصفر.

يجب الانتباه إلى أن تعريف هذا السجل يتم خارج الحلقة وليس داخلها وإلا سيتم إعادة السجل إلى القيمة البدائية في كل مرة.

بعد ذلك سيتم سحب القيمة الموجودة في عنوان si وجمعها مع القيمة الموجودة في عنوان bx ونقل الناتج إلى عنوان di. وبالتالي سيتم تخزين القيمة الأولى. سيتم بعد ذلك زيادة العناوين للانتقال للقيمة التالية وإجراء الجمع حتى الانتهاء من عمليات التكرار.

وظيفة:

اكتب إجراء بلغة الأسمبلي يقوم بجمع عناصر مصفوفة مؤلفة من العناصر 4 ،2 ،2 ،1 ومن ثم تخزين الناتج في موقع ما بالذاكرة اسمه res.

تمرين3:

اكتب برنامج يقوم باستخدام الإجرائيات من أجل تعبئة 32 قيمة بدءاً من العنوان 200H ومن ثم نقل تلك القيم إلى العناوين بدءاً من 240H. اعتمد في عملية التعبئة على الإجرائية FILL وفي عملية النقل على الإجرائية MOVE.

الحل:

سيبدأ البرنامج باستدعاء الإجراء CALL والذي ستتم العودة منه مباشرة بعد تنفيذه ليتم الانتقال إلى السطر التالي والذي يمثل استدعاء الإجراء MOVE. ومن ثم سيتم طلب المقاطعة لإنهاء البرنامج.

```
ORG 100H

CALL FILL

CALL  MOVE

INT 5H

;

FILL:

        CLD;  Reset Dirction Flag

        MOV SI, 0200H

        MOV DI, 0200H

        MOV AL, 0CDH

        MOV CX,20H

        REP STOSW

        RET

MOVE:   MOV SI, 0200H

        MOV DI, 0240H

        MOV CX,20H

        REP MOVSW

        RET
```

في هذا القسم قمنا بتحميل موقع أول عنوان نرغب بالتخزين ضمنه إلى السجلين SI,DI حيث يمثل السجل SI السجل المصدر في حين يمثل المسجل DI المسجل الهدف. وتم شحن السجل al بالقيمة 0CD وتم تحميل السجل CX المخصص للتحكم بالحلقات (عدد التكرارات المطلوب أن يتم التخزين وفقاً لها) بالقيمة 20H وهي تقابل القيمة32 بالعشري. تسمح التعليمة STOSW بتخزين القيمة في الموقع المحدد من الذاكرة ومن خلال تعليمة التكرار REP ستتكرر المهمة مع إنقاص السجل cx (وهو سجل العداد الذي تتعلق به عملية النقل والتخزين) وعند وصول هذا السجل إلى الصفر ستتوقف عملية التكرار.

سيتم في هذا القسم تحميل المسجل المصدر بالعنوان المطلوب النقل منه وبالمسجل الهدف العنوان الذي نريد النقل إليه ومن ثم سيتم تحديد عدد مرات التكرار أيضاً بمسجل العداد وسيتم القيام بنقل القيمة بين العناوين بصورة متسلسلة وفقاً للتعليمة MOVSW والتي تتعلق أيضاً بسجل العداد

تمرين4:

اكتب برنامج يقوم بتخزين الأعداد من 1 إلى 10 ضمن مجال العناوين بدءاً من 200H.

الحل:

سنعتمد هنا على المسجل AL ليقوم بمهمة العداد حيث يبدأ من القيمة 0 ويستمر بالتزايد حتى الوصول للقيمة 10. في حين سيتم شحن قيمة العنوان المصدر والذي نرغب بالتخزين ضمنه في المسجل SI.

```
MOV AL,00H

mov si, 0200h

FILL:

MOV [SI],AL

INC AL

INC SI

CMP AL,10H

JNZ FILL
```

يبدأ البرنامج بعد ذلك بحلقة تكرارية حيث يتم نقل محتوى المسجل AL إلى العنوان المحدد ومن ثم زيادة قيمة العدد الموجود في المسجل AL (أي كانت القيمة 0 ستصبح بعد تعليمة INC بمقدار 1) كما سيتم في الخطوة التالية زيادة العنوان (كان 200 وسيصبح 201) ومن ثم سنقارن قيمة السجل AL والذي يمثل القيمة المتزايدة مع القيمة 10 فإذا كانت تساويها يتم الخروج من الحلقة وإلا سيتم تكرار عملية النقل بين السجلات.

تمرين5:

يطلب كتابة برنامج يقوم بحساب محيط دائرة باستخدام الموجهات حيث سيتم تخزين قيمة PI=3.14 في موجه أول ويتم تخزين نصف القطر D=2 في موجه ثاني ومن ثم تخزين الخرج في موجه ثالث. يطلب أن يكون نمط تخزين البيانات ككلمة word.

الحل:

```
ORG 100h

.DATA

PI DW  0003H

DI DW  0002H

RES DW ?

.CODE

MOV AX, VAR_1

MOV BX, VAR_2

MUL BX

MOV AX,BX

MOV BX, 0002H

MUL BX

MOV RES,BX

RET
```

تمرين 6:

يطلب كتابة إجراء برمجي يقوم بتحديد أكبر عدد ضمن مصفوفة من الأعداد 08h,14h,05h,0Fh,09h اعتماداً على الموجهات على أن يتم تخزين القيمة الأكبر ضمن موضع محجوز في الذاكرة باسم res.

الحل:

سنقوم في البداية بتعريف موجه ونعطيه الاسم STRING1 ونقوم بتخزين اتلقيم المعطاة للمصفوفة, كما سنقوم بتخزين الموقع الهدف بالاسم res.

```
data segment

STRING1 DB 08h,14h,05h,0Fh,09h

res db ?

data ends

code segment

assume cs:code, ds:data
```

سنقوم في المرحلة التالية بنقل قطاع البيانات إلى المسجل AX وتحميل المسجل CX بعدد عمليات المقارنة في حين تم تحميل المسجل bl بالقيمة 0 وهي القيمة الأولى التي سنقوم بالمقارنة معها.

أي أن مخطط سير البرنامج يتضمن مقارنة أول قيمة في المصفوفة مع القيمة المخزنة في المسجل bl (وهي في الوضع الحالي تساوي الصفر) فإذا كانت القيمة التي في المصفوفة أكبر سيتم الاستبدال وإلا فإننا نحافظ على القيمة الموجودة في المسجل bl وتتكرر العملية إلى أن يتم الانتهاء من العناصر الموجودة في المصفوفة.

```
start: mov ax, data

mov ds, ax
```

```
mov cx, 04h
mov bl, 00h
LEA SI, STRING1
up:
mov al, [SI]
cmp al, bl
jl nxt
mov bl, al
nxt:
inc si
dec cx
jnz up

mov res,bl
end start
```

تحميل عنوان أول قيمة تابعة للمصفوفة إلى المسجل SI أي سيحتوي هذا المسجل على عنوان القيمة 08h في المصفوفة

تم بعد ذلك نقل القيمة الموجودة في العنوان المحدد وهي 08h في المصفوفة السابقة إلى المسجل al

سيتم بعد ذلك المقارنة بين المسجلين al,bl , تم اختيار شرط المقارنة للقيمة الأصغر باستخدام التعليمة jl فإذا كان المسجل al أصغر من المسجل bl سيتم القفز إلى nxt وإلا سيتم تنفيذ التعليمة التالية ونقل al والذي هو أكبر إلى bl

ضمن هذه الحلقة nxt سيتم زيادة si للانتقال إلى القيمة التالية في المصفوفة وانقاص عدد المقارنات ضمن سجل العداد وفي حال لم يساوي cx القيمة صفر يتم العودة إلى إجرائية up ومقارنة محتوى السجل al الجديدة مع bl.

بعد الانتهاء من جميع التكرارات سيتم تخزين العدد الأكبر ضمن المسجل bl وسيتم نقله بعد ذلك إلى المسجل المحدد في الذاكرة res.