



أنماط العنونة الأربعة

الأهداف التعليمية:

سنتعلم في هذا الفصل:

1. لماذا يمتلك الحاسوب العديد من أنماط العنونة.
 2. تطبيقات العنونة الفورية.
 3. العنونة غير المباشرة وتطبيقاتها.
 4. العنونة النسبية للحاسوب والمستخدم من أجل القفز إلى التعليمات القريبة.
 5. العنونة الدلالية والقاعدية وكيفية استخدامها في عنونة عناصر مصفوفات متعددة الأبعاد.
 6. استخدام العنونة القاعدية في عملية إعادة توضع البرنامج.
-

هناك العديد من أنماط العنونة التي يمكن استخدامها عند تصميم المعالجات. سندرس في هذا الفصل أربعة أنماط مشهورة للعنونة ومستخدمتها حواسيب شركة IBM. نبدأ على مصمم وحدة المعالجة المركزية أن يضيف مسجلات خاصة وتعليمات آلة جديدة لتدعم أنماط العنونة هذه.

4.1 ما هي الحاجة لأنماط العنونة

هناك حاجة أي برنامج حاسوبي لعنونة المعطيات والتعليمات. ففي المثال الذي أعطى الشكل (3-1) الفصل الثالث، تعنون تعليمة `MOV CX, 3` (ثابت بقيمة 3)، وتعنون تعليمة `LOOP` تعليمة أخرى مسبقة باللائحة `next`، ومثل هذه العناوين هي جزء من التعليمة. هناك بعض التعليمات مثل `JMP` و `LOOP` تاج لعنوان واحد وهناك تعليمات أخرى مثل `MOV AX, num` تاج لعنوان، المعنون اسجل وعنوان المعطيات المخزنة في الذاكرة RAM.

يؤثر طول العنوان بشكل مباشر على طول التعليمات. تؤثر بدورها على حجم الذاكرة RAM المطلوب لتخزين البرنامج.

إن أبسط ط للعنونة يعرف بالعنونة مباشرة (direct addressing)، وهذا النمط يحتاج إلى عنوان بطول n بت لعنونة 2^n موقعاً في الذاكرة RAM. ومع زيادة تعقيد البرنامج وتوسع الآلات استخدامها ازداد الطلب على ساعات أكثر للذاكرة، فكمي نقوم بعنونة مباشرة للذاكرة RAM بسعة 64MB نحتاج لعنوان بطول 26 بتاً ($2^{26}=64M$). إن استخدام طريقة تشفير وأسطح عنونة ذكية سيؤدي إلى خفض طول هذا العنوان بشكل ملحوظ. إن نقص طول العنوان سيكون على حساب زمن الوصول إلى المعطيات في الذاكرة كما سنرى لاحقاً.

إن عدد المسجلات العامة بشكل عام قليل وهو من رتبة 8، 16، 32، 64 وهذه المسجلات قابلة للعنونة مباشرة، كما إن حقول العنوان للمسجل بالتعليمات يشكل طولاً ثابتاً من رتبة 3، 4، 5، 6 بت ولطالما أن المسجلات موجودة ضمن المعالج فإنه يتم الوصول إليها بشكل أسرع من الوصول إلى مواقع الذاكرة، ولابد من لغة التجميع من استخدام المسجلات قدر استطاع ليقفل من زمن تنفيذ البرنامج.

لا تنفيذ أسطح العنونة تقليل كمية التخزين للبرنامج فحسب وإنما تنفيذ أيضاً تقليل زمن تنفيذ البرنامج وهذا ما سنراه في الفقرة التالية. بالإضافة إلى الفعالية التخزين و زمن تنفيذ البرنامج فإن هناك مفهوم نوعي مرغوب خاص بالبرنامج والمعمارية وقابلية القراءة والكتابة يساعدان صيانة البرنامج، لاحظ التعليمات التالية:

ADD AX, BX (instruction-1)

هذه التعليمات تستخدم العنونة مباشرة لعملية التعليمات ADD (المصدر والوجهة)، وهذان العاملان يشيران إلى المسجلات. من جهة أخرى إن التعليمات التالية:

ADD A, B (instruction-2)

تستخدم العنونة مباشرة وكلتا المعاملات موجودان في الذاكرة RAM. من الواضح بأن التعليمات (instruction-2) ستكون أكثر طولاً من التعليمات (instruction-1). فإذا كان لابد من استخدام 26 بتاً لعنونة ذاكرة RAM بسعة 64MB بشكل مباشر فإنه من الواضح بأن حقول المعاملات للتعليمات (instruction-2) سيحتاج إلى 52 بتاً لوحده، وهذا يعني طويلاً جداً وذلك عندما نرغب في التعليمات بطول 32 أو 48 بتاً. الحل البديل هو أن نستخدم مسجل من أجل إحدى المعاملات ونستخدم العنونة مباشرة في الذاكرة RAM من أجل المعامل الآخر:

ADD AX, Y

وهناك طرق عنونة أخرى تستخدم لتقليل طول التعليمات وهذا ما سنناقشه في قطع 4.4 و قطع 4.5.

4.2 العنوان الفورية

إذا كانت إعطيات المطلوب استخدامها من قبل التعليمة تشكل جزءاً من هذه التعليمة فإنه زمن التنفيذ، عند جلب هذه التعليمة من الذاكرة، سيتم جلب إعطيات مطلوبة معها، وهكذا لا توجد حاجة لأي زمن إضافي لمعالجة هذه الإعطيات. هذا النمط من عنوان إعطيات يدعى بالعنوان الفورية (immediate addressing)، انظر المثال التالي :

MOV CX, 7

إن إعطيات هذه الكلمة هي ثابت بدون إشارة مساوٍ للقيمة 7 وهو يشكل جزءاً من التعليمة. فإذا كان طول التعليمة 16 بتاً منها 8 بتات مخصصة لشيفرة العملية (MOV CX) فإن الثمان بتات الباقية ستخصص للمعطيات. وهكذا ثمان بتات تصل على 256 تركيبة يمكن استخدامها لتمثيل الأرقام الصحيحة بدون إشارة من صفر و+255. يمكن بشكل بديل أن تعامل هذه الثمان بتات على أنها أعداد بإشارة عن طريق استخدام التناهي و- هذه الكلمة سيكون الـ 128 و+127.

إذاً، فأحد مددات العنوان الفورية هو أن حجم المعامل صغ، وبشكل طبيعي يتم استخدام 8 بتات لتمثيل معاملات العنوان الفورية.

تنفذ التعليمات التي تستخدم العنوان الفورية بشكل سريع وهذا عائد لكون إعطيات لمعالجة التعليمة. وبصرف النظر عن كون حجم إعطيات دون فإنه توجد سلبية أخرى استخدام العنوان الفورية. لنفترض أننا نقوم بتجميع برنامج يوي التعليمة MOV CX, 7 من أجل تطبيق مع وبعد فترة ما أردنا أن نغير القيمة 7 لتصبح 3، هذا يعني أنه يجب أن نغير التعليمة لتصبح MOV CX, 3 ونعيد جميع البرنامج. وهذا لن يكون قضية أساسية عند العمل مع برامج صغيرة، ولكن مرونة البرنامج تصبح مفقودة هنا.

الإمكانيات الأخرى للعنوان الفورية أوسع من إمكانية قراءة البرنامج، ففي سلسلة المعالجات 80×86 مصص المسجل CX كعداد مصص لتعليمات LOOP، فعندما يرى القارئ التعليمة MOV CX, 7 متبوعة بتعليمة LOOP سيفهم القارئ بسهولة بأن الحلقة ستتكرر سبع مرات.

عند تنفيذ العبارة SUM = SUM + X بشكل متكرر حلقة حساب القيمة التراكمية للمتغير X فمن الأساسيات أن تؤسس قيمة المتغير SUM في الصفر قبل بدء تنفيذ الحلقة، حيث إن طبيعة العنوان الفورية مناسب جداً للتأسيس. تلك لغة التجميع العديد من الميزات المتقدمة لاستخدام العنوان الفورية بشكل فعال، وإحدى هذه الميزات معطاة من خلال التوجيه اسمي EQU. تسمح هذه التعليمة للمبرمج باستخدام اسم رمزي لثابت ومن يري استخدامه

العديد من المواقع ال نامج، وسيقوم ا مع بالتسوية ما ب الاسم الرمزي والثابت اقابل ويستبدل الاسم بالثابت خلال عملية التجميع، كما اثال التا :

```
Maxint      EQU    32767 ;
.
.
MOV    BX, maxint ;
.
.
MOV    DI, maxint ;
.
.
```

يتم تعي القيمة 32767 كل من اسجلا BX وDI. عند تعي القيمة Maxint فإن ا مج اجة فقط لتعي ها مكان واحد فقط وذلك عبارة EQU.

كن أن تكون اعطيات الفورية قيمة صحيحة أو رف آسكي، حيث يتم وضع ا رف ضمن علام اقتباس مفردة كما هو م :

```
MOV AX, 'a'
```

4.3 العنونة المسجلية غير المباشرة

تستخدم تويات اسجلات (BX على سبيل اثال) كمعطيات أو كعنوان (مشة إ موقع الذاكرة RAM) لبعض اعطيات اوجودة الذاكرة. يسمى الشكل الأول بالعنونة اسجلية اباشرة ويسمى الشكل التا بالعنونة اسجلية غ اباشرة. انظر إ اثال ا ب الشكل (4-1)، إن اواقع الذاكرة من 200 و 207 توي على سلسلة من ا ارف وهي GOOD-DAY. سوف ننفذ التعليمات التالية:

```
MOV BX, 200 ;          200 هي العنوان القاعدي
ADD BX, 6 ;             توي BX الآن على القيمة 206
MOV DL, [BX] ;          [ ] يستخدمان للدلالة على العنونة اسجلية غ اباشرة
```

	G	O	O	D	-	D	A	Y
Address: →	200	201	202	203	204	205	206	207

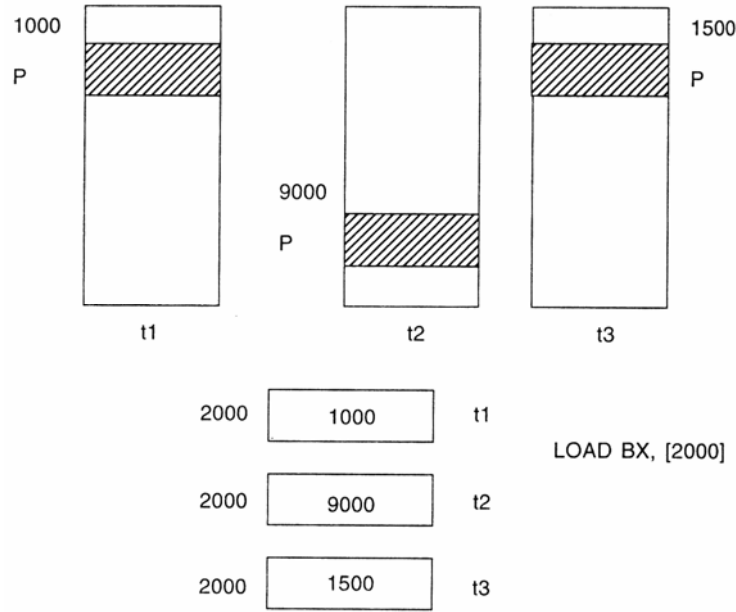
الشكل (4-1): محتويات الذاكرة RAM

بعد تنفيذ التعليمات الثلاث السابقة سيحتوي سجل DL على توابات موقع الذاكري ذي العنوان 206 و AOI على A ، إن الأقواس أربعة [] أيطلة بالرمز BX تستخدم لإعلام مع بأن العنوان المستخدمة هي عنوان مسجلة $\bar{G}$ مباشرة. فإذا كتابة $MOV DL, BX$ بدلاً من $MOV DL, [BX]$ فإنه سيتم زين الرقم 206. سجل DL. إن التعليم $MOV DL, BX$ $\bar{G}$ صحيحة قواعدياً حيث أن BX مسجل بطول 16 بتاً و DL مسجل بطول 8 بتات.

يتكون سجل BX العائلة 80×86 من 16 بتاً وبإمكانه عنوان 2^{16} (64K) موقعاً الذاكرة RAM. واعتماداً على الوجهة لتعليم MOV (وال كمن أن تكون 8 بتات أو 16 بتاً أو 32 بتاً اعطيات بينتيوم) فإنه بالإمكان استخدام معطيات بطول أكبر من 8 بتات. وهكذا فإن حجم اعطيات و ال العنوان كمن زياد ما من خلال استخدام العنوان اسجلية $\bar{G}$ مباشرة. خلال تنفيذ التعليم $MOV DL, [BX]$ ، يتم الدخول إلى الذاكرة لمعامل الذي سيتم نقله وبالتالي ستكون هذه التعليم أبطأ من التعليم ال تستخدم مسجلاً وعنوان فورية.

إن العنوان $\bar{G}$ مباشرة $\bar{G}$ مقيدة استخدام اسجلات فقط، إذ يوجد أيضاً عنوان $\bar{G}$ مباشرة تعتمد على الذاكرة، وهذه الة يتم الرجوع إلى موقع الذاكري الذي سيحتوي على عنوان اعطيات وليس اعطيات مطلوبة. عند استخدام هذا النوع من العنوان سيزداد زمن التنفيذ لأنه من الواجب الدخول للذاكرة RAM مرة ، مرة لمعامل و مرة الأخرى لمعامل اعطيات من هذا العنوان.

سنوضح كيفية استخدام العنوان $\bar{G}$ مباشرة من خلال أمثلة الشكل (2-4) والذي يبين وضعية الذاكرة خلال ثلاث فترات زمنية متلاحقة t_1, t_2, t_3 . لاحظ بأن ال نامج (+ اعطيات) P أعيد توزيعها ديناميكياً بدءاً من العناوين 1000، 9000، 1500 على التنا وهذا كمن أن حدث كنتيجة للعديد من الأسباب. عند إعادة ميل P فإن عنوان القاعدة سوف يظل التغييرات. و الشكل السابق اختيار العنوان 2000 يث أن عنوان القاعدة المتغ ديناميكياً كمن ميله من قبل نظام التشغيل. يستطيع ال ميج من خلال العنوان $\bar{G}$ مباشرة ميل عنوان القاعدة ال سجل BX من خلال التعليم $MOV BX, [2000]$. لاحظ بأنه لا توجد مشكلة عند مكان وجود P حيث أنه سيتم ميل العنوان القاعدي سجل BX من خلال هذه التعليم ويكون نظام التشغيل قادراً على استبدال العنوان 2000 بالعنوان القاعدي ال . إن عملية إعادة التوزيع الديناميكية لل نامج P لا تغ التعليم MOV ، وهذا السبب حالاً يتم مع ال نامج الذي توي على هذه التعليم فإنه من $\bar{G}$ المطلوب إعادة معيه. لقد هذا أمثلة استخدام العنوان الذاكري $\bar{G}$ مباشرة. سيكون من استحسن استخدام العنوان اسجلية $\bar{G}$ مباشرة وذلك عند توفر عدد كاف من اسجلات وحدة المعالجة المركزية.



الشكل (4-2): استخدام العنونة غير المباشرة.

4.4 العنونة النسبية

لاحظ أسئلة التالية: اقرأ برؤ من لوحة المفاتيح (برؤ كل مرة) فارما، فإذا كانا متساويين أكتب Equal وإلا أكتب Not Equal على الشاشة، كرر هذه العملية ثلاث مرات.

إن برنامج التجميع الذي ينجز هذه الوظيفة مبين الشكل (3-1). تغير هذا البرنامج بشكل بسيط ليبدو كما الشكل (4-3). يتم زبن أول بايت مقروء الذاكرة RAM موقع اسمي temp، و زبن البايت الثاني المسجل AL، حيث تتركز التعليمة INT 21h البايت المقروء من لوحة المفاتيح. الخطوة الثانية البرنامج هي مقارنة هذين البايت وعرض إما كلمة Equal أو كلمة Not Equal فإذا كان البايتان متساويين فإنه سيتم تنفيذ الأسطر 7، 8، 13، 14، 15 وإذا كانا مختلفين سيتم تنفيذ الأسطر 7، 8، 9، 10، 11، 12. تتكرر هذه العملية ثلاث مرات من خلال تعليمة Loop. السطر 8 توجد التعليمة JE Equal و السطر 12 التعليمة JMP Next، وكلتا هاتين التعليمتين تستخدمان لافتات عنونة من أجل التفرع.

سوف نستخدم المصطلح "عنوان التعليمة الذا" من أجل موضع التعليمة ونشبه إليه بالرمز *. نلاحظ تعليمات القفز السابقة أن لافتات العناوين الفرعية قريبة من تعليمات القفز نفسها.

إذا كان الرمز * هو عنوان التعليمة JE Equal (السطر 8) فإن $+10 * +10$ إذا افترضنا أن كل من التعليمات المسماة بطول 2 بايت) سيعطي عنوان الالفة Equal السطر رقم 13. إن القيمة الرقمية للرمز * هي نفسها تبويات عداد ال نامج PC عند الوصول هذه التعليمة. وبشكل مشابه لكي يصل على عنوان الالفة next (السطر رقم 16) من تعليمة JMP Next (السطر 12) فنحن نحتاج إلى $+8 *$ وذلك بفرض أن التعليمات الأربع التالية إلى تتبع السطر 12 بطول 2 بايت. هناك تعليمات ددة لغة التجميع للمعامات 80×86 وهي ليست بطول 2 بايت. سوف يستخدم مع الطول الصحيح لكل تعليمة وذلك سبب العنوان النسبي. يمكن للعنونة النسبية أن تكون تفرعاً للأمام (positive) + وقد تكون تفرعاً للخلف (negative) -. إن لالفة العنونة firstchar أشار إليها تعليمة loop السطر 16 توجد السطر رقم 2 وهذه الالة سيكون العنوان النسبي -28. وهكذا فإن القيمة النسبية اللازمة للوصول إلى هذه الالفة لابد أن تكون عدداً صحيحاً يمكن أن يأخذ قيمة موجبة أو سالبة، وهذا الثابت زن كجزء من تعليمة التفرع.

يستخدم عائلة المعامات 80×86 انية بتات لتخزين العنوان النسبي ، فعند معاملتها على أ عدد بإشارة من خلال استخدام اتمم الثنائي فإن هذه البتات الثمان ستضم الّا من -128 و+127. ونؤكد هنا أن القيمة الرقمية للعنوان النسبي لا سبب من قبل ا مج و ا من قبل برنامج ا مع حيث ينبغي على ا مج أن يش فقط إلى مدى بعد التعليمات ال سيتم القفز إليها عن التعليمة الالية. فعندما تقع العناوين الفرعية ضمن الال (PC-value-128) إلى (PC-value+127) فإنها تسمى لالفة قصه SHORT، والتقليد استخدم من قبل معامات شركة مايكروسوفت للدلالة على هذا التفرع يأخذ الشكل التالى :

```
JMP SHORT label;
```

عندما لا تذكر SHORT فإنه يتم اعتبار الالفة label قصه. أما الصفات الأخرى الممكنة للالفة هي NEAR PTR وFAR PTR وال ستتم مناقشتها لاحقاً هذا الكتاب.

	.data		
	temp	DB	' '
	eql	DB	'Equal', '\$' ;
	noteql	DB	'Not-Equal', '\$' ;
	.code		
1		MOV CX 3h	; To DO 3 times
2	firstchar:	MOV AH, 01h	; code for Read
3		INT 21h	; First Byte
4		MOV temp, AL	; save in temp
5		MOV AH, 01h	;
6		INT 21h	; For second byte

```

7          CMP AL, temp          ;
8          JE equal
9          MOV AH, 9h            ; code for output
10         MOV DX, offset noteql ;
11         INT 21h
12         JMP next
13  equal:  MOV AH, 9h            ;
14         MOV DX, offset eql;
15         INT 21h
16  next:   LOOP firstchar        ; next iteration
17         INT 20h                ; end program execution

```

الشكل (4-3): برنامج لمقارنة محرفين

4.5 العنونة الموجهة بالإزاحة

تنتمي العنونة الدليلية (indexed addressing) والعنونة القاعدية (base addressing) إلى هذا النمط من العنونة، حيث يوجد عنوان مرجعي وهو ما يسمى بالعنوان القاعدي (Base address). تسمى إضافة ما به هذا العنوان القاعدي والعنوان المطلوب الوصول إليه بالإزاحة (displacement). تضاف هذه الإزاحة إلى العنوان القاعدي للحصول على ما يسمى "بالعنوان الفعال" EA (effective address)، حيث يشي العنوان الفعال إلى موقع زين التعليمة أو إعطيات المطلوبة.

على الرغم من أن العنونة الدليلية والعنونة القاعدية تستخدمان نفس القاعدة للحصول على العنوان الفعال EA إلا أنهما تستخدمان لأغراض برية مختلفة كلياً. يوجد عائلة معالجات 80×86 نوعاً من المسجلات المسجلة القاعدة (base registers) BX وMSJلات الدليل DI, SI وMSJلات (index register). يستخدم مسجل الدليل مع مصفوفات الأرقام ومصفوفات الأرف، إذ يمكن أن تكون هذه المصفوفات ببعد وحيد (شعاع) ويمكن أن تكون ببعدين (مصفوفة). عند استخدام العنونة الدليلية (الفهرسة) فإن العنوان القاعدي سيكون عنوان أول عنصر المصفوفة وسيتم وضع الإزاحة مسجل الدليل، ومن خلال زيادة تويات مسجل الدليل يمكن تغيير قيمة العنوان الفعال EA لعنونة كل عنصر المصفوفة. تفيد كل من تعليمة INC (زيادة توى مسجل) وتعليمة Loop عمليات التعامل مع المصفوفات.

من جهة أخرى، تفيد العنونة القاعدية عملية إعادة توضع المجمع أو إعطيات الذاكرة، حيث رأينا في الفصل الثالث الحاجة لإعادة توضع المجمع أنظمة التشغيل ذات المة المتعددة. فعند إعادة توضع المجمع الذاكرة سيتم زين العنوان القاعدي الجديد مسجل القاعدة.

لا يوجد | واسب 80×86 عدد كاف من المسجلات لكي يصص مسجلاً بشكل دائم من أجل العنونة القاعدية، وعلى كل حال، فإن طول التعليمات | عا مات 80×86 سيبقى صفراً طالما | تستخدم العنونة القاعدية مع كل تعليمة. فخلال التعامل مع مصفوفة ببعدين يتم استخدام مسجل القاعدة من أجل عملية الفهرسة، وهذا مب | الشكل (4-4) حيث تعريف مصفوفة بثلاثة أسطر وأربعة أعمدة | ينالها miles. وللتوضيح فقط افترضنا أن العنوان القاعدي للمصفوفة miles هو 200 وأن كل عنصر | صفوفة يشغل بايتاً واحداً | الذاكرة RAM.

Address	200	201	202	203	204	205	206	207	208	209	210	211
miles	18	28	38	48	16	26	36	46	14	14	34	44
	row 1				row 2				row 3			

تشير إلى آخر عنصر وهو 44 : $[2 \cdot BX + SI] = 208 + 3 = 211$

الشكل (4-4): تخزين مصفوفة في الذاكرة

انظر | السطر رقم 4 | الشكل (4-5)، فمن خلال وضع القيمة 4 (عدد أعمدة | صفوفة) | سجل BX | صل على عنوان فعال من خلال التعب | miles $[2 \cdot BX]$ | العنصر الأول من السطر الثالث للمصفوفة miles | ساوي | 208. | توي مسجل الدليل على القيمة 3 كما هو مب | السطر 5.

1	miles	DB	18,	28,	38,	48	;	row	-1
2		DB	16,	26,	36,	46	;	row	-2
3		DB	14,	24,	34,	44	;	row	-3
		.							
		.							
		.							
		MOV					;		
4		MOV	BX,	4			;		
5		MOV	SI,	3			;		
6		MOV	AL, miles		$[2 \cdot BX + SI];$				
		.							
		.							
		.							

الشكل (4-5): شيفرة برمجية لعنونة عنصر في مصفوفة.

يتم | السطر رقم 6 | دمج كل من العنونة الدليلية والعنونة القاعدية مع بعضها كقيمة إزاحة | اعتباراً من العنوان 200، ليصبح العنوان الفعال $208 + 3$ | ساوي | 211. | تذكر أن الرمز miles يقابل القيمة 200 | هي العنوان القاعدي للمصفوفة.

EA = Effective Address = 2 * contents of BX + contents of SI + displacement value
= 8+3+200 = 211 (in the example discussed)

لوصول إلى العنصر (i, j) هذه الصفوفة، أي العنصر الموجود السطر رقم i والعمود رقم j،
نستخدم العبارة miles [200+(i-1)*4+(j-1)]

يشي العنوان الفعال للمثال المذكور سابقاً إلى العنصر الرابع السطر الثالث، أي موقع 211 من الذاكرة و هو على القيمة 44. فمن خلال زيادة BX بأربعة 4 كل مرة نستطيع اختيار أسطر تلقية، ومن خلال زيادة SI بواحد 1 نستطيع اختيار عمود السطر اختار. وهكذا فإن العنونة ذات البعدين لأسطر وأعمدة الصفوفة miles قد حولتها إلى عناوين خطي الذاكرة RAM.

سنوضح الآن من خلال برنامج آخر عملية الفهرسة ببعد واحد.

مسألة: مصفوفة بها seq بسبعة عناصر كل منها بطول بايت واحد. زنة الذاكرة RAM. ب على الـ نامج أن يفحص ما إذا كانت هذه الصفوفة مرتبة حسب الحروف الأدية، فإذا كانت مرتبة يعرض حرف T (true) وإلا فإنه سيعرض F (false). سنستخدم العنونة الدليلية (الفهرسة) لـ هذه المسألة. تذكر أن شيفرة الأسكي للحروف من A و Z مرتبة تصاعدياً، والـ نامج الذي يقوم بذلك معطى الشكل (4-6).

```
.data
seq DB 'ADEPSWZ' ;
.code
MOV CX, 6 ; Loop count (n-1) where n
; is the array size
MOV SI, 0 ; initialize index to 0
next: MOV AL, seq [SI] ; load ith value of seq in AL
MOV BL, seq [SI+1] ; load (i+1) value of seq in BL
CMP AL, BL ; compare characters
JG notseq ; JG is jump if greater
INC SI
LOOP, next ; LOOP back to check next
; element
MOV AH, 2 ; code for display
MOV DL, 'T' ; T for True
INT 21h ; Display T
INT 20h ; Halt execution
notseq: MOV AH, 2 ;
MOV DL, 'F' ; F for false
INT 21h ; Display F
INT 20h ; Halt execution
```

الشكل (4-6): برنامج لفحص الترتيب التصاعدي للبايتات

تصميم البرنامج

قبل كتابة البرنامج لابد من تحليل المشكلة، وتصميم الخوارزمية وبعدي التخطيط لكيفية تطوير شيفرة لغة التجميع. إنه لعمل جيد أن تقوم بتوثيق ذلك بوضوح، وهذه الوثائق يجب إعادة كتابتها إذا غابنا الشيفرة مرة أخرى.

سنقدم الآن مثالا يبين كيفية توثيق البرنامج المعطى بالشكل (4-6):

- تسمى مصفوفة المعطيات بالاسم seq وهي مصفوفة ببعد واحد ووي سبعة عناصر كل منها بطول بايت واحد.
 - نقارن القيمة رقم i-th من المصفوفة مع القيمة ذات الرقم (i+1) فإذا كانت القيمة ذات الرقم i-th أكبر من القيمة ذات الرقم (i+1) فإن المصفوفة مرتبة وبالتالي نوقف الملقنة ونعرض كلمة False يتوقف البرنامج.
 - إذا انتهت حلقة المقارنة بشكل طبيعي فهذا يعني أن المصفوفة مرتبة بشكل صحيح.
 - إن تعليمة المقارنة CMP تسمح رجعية وحيدة في الذاكرة RAM ولذلك يتم نقل العنصر رقم i-th إلى سجل AL.
 - يتم تكرار حلقة المقارنة (n-1) مرة، حيث أن n مثل عدد العناصر في المصفوفة.
- إن البرنامج المبين بالشكل (4-6) مبني على التصميم المذكور وهو يوضح نفسه بنفسه (انظر الملحق 3 من أجل الشيفرة الكاملة).

4.6 ملخص

جرى هذا الفصل توضيح أربعة أنماط عنونة أحدها (العنونة الموجهة بالإزاحة) مثل نوعاً عاماً ويوضح نوعاً للعنونة المباشرة القاعدية والعنونة الدلالية. تعد العنونة المباشرة للمواقع الذاكرة أو للمسجلات إحدى طرق العنونة المشهورة ونعتيها إحدى طرق العنونة هذا الفصل. إن التعليمة MOV AX, BX على سبيل المثال توظف عنونة مباشرة مسجلة، والتعليمة MOV AX, SUM توظف عنونة مباشرة ذاكرية.

تفيد أنماط العنونة جميع طرق عديدة، فهي تساعد كتابة برنامج قابل للقراءة والصيانة بسهولة، كما أنها تساعد استخدام العنونة بدون ميل لزيادة كفاءة على طول التعليمة. سنأناط العنونة (العنونة الفورية) بعض الحالات من زمن التنفيذ وذلك على حساب البرونة، وأحياناً أخرى (العنونة المباشرة) سنأناط العنونة على حساب زيادة زمن التنفيذ.

تمارين

ضع الإجابة الصحيحة الفراغ :

- 4.1 يستخدم ط العنوان مباشرة عندما
 - (a) تكون السرعة هامة
 - (b) يطلب حجم عنوان كبير
 - (c) يطلب كتابة برنامج معقد
 - (d) يطلب كل ما ذكر
- 4.2 لدينا حاسوباً بذاكرة 32 MB. يلزم بتاً حقل العنوان للتعليمية وذلك لعنوان هذه الذاكرة بعنوان مباشرة.
 - (a) 16
 - (b) 20
 - (c) 32
 - (d) 25
- 4.3 يزداد الحجم الذي يشغله البرنامج عندما:
 - (a) تتطلب صيغة التعليمات بتات أكثر.
 - (b) يلزم معالجة مصفوفات معطيات حجم كبير.
 - (c) كلا السببين المذكورين.
- 4.4 تعنون المسجلات بشكل مباشر عندما:
 - (a) تزداد سرعة المعالجة
 - (b) يقل عدد المسجلات
 - (c) يكون طول المسجلات 16 بتاً
 - (d) لا شيء من هذا.
- 4.5 الزمن اللازم لمعالجة معطيات ط العنوان الفورية يساوي الصفر.
 - صح
 - خطأ
- 4.6 عدد البتات اللازم لعنوان فورية مساوية ± 1024 هو بت.
 - (a) 10
 - (b) 11
 - (c) 12
 - (d) 20
- 4.7 إحدى ميزات العنوان الفورية هي.....
 - (a) الوضوح خلال قراءة البرنامج
 - (b) سهولة فك تشفير المعاملات
 - (c) التنفيذ الفوري للتعليمية
 - (d) لا حاجة لمعامل من الذاكرة

- 4.8 عدد مرات الدخول إلى الذاكرة المطلوب لمعامل الفوري يساوي.....
 (a) 0 (b) 1 (c) 2 (d) 3
- 4.9 سيئة العنونة الفورية هي:
 (a) عدم إمكانية استخدام الأعداد السالبة
 (b) ضرورة إعادة ميع الـ نامج عند تغير قيمة المعامل الفوري
 (c) ضعف وضوح الـ نامج
- 4.10 كن للمعامل الفوري أن يكون راف أسكي
 • صح • خطأ
- 4.11 تتطلب عنونة الذاكرة غباشرة..... دخولاً إلى الذاكرة لمعطيات
 (a) 0 (b) 1 (c) 2 (d) 3
- 4.12 تاج العنونة اسجلية غباشرة..... دخولاً إلى الذاكرة لمعطيات
 (a) 0 (b) 1 (c) 2 (d) 3
- 4.13 عدد اواقع امكن عنونتها باستخدام العنونة اسجلية غباشرة اواسب الشخصية يساوي..... موقعاً.
 (a) 256 (b) 1024 (c) 64K (d) 256K
- 4.14 يفيد ط العنونة..... إعادة توضيع اعطيات والامج الذاكرة.
 (a) باشرة (b) غباشرة (c) الفورية (d) النسبية (PC-relative)
- 4.15 العنونة النسبية تعتمد على اسجل كعنوان مرجعي.
 (a) DX (b) DS (c) IP (d) SI
- 4.16 تستخدم العنونة النسبية كل من الأعداد السالبة واجبة.....
 (a) لتكون عامة بطبيعتها
 (b) لعنونة كل من التعليمات واعطيات
 (c) للتفرع للأمام واللف.
- 4.17 تستخدم..... بتات سلسلة اعالمات 80×86 لتخزين العنوان النسبي .
 (a) 16 (b) 8 (c) 32 (d) 10

- 4.18 تستخدم العنونة النسبية بشكل أو التعليمات.....
- (a) ا سابية (b) القفز (c) النقل
- 4.19 كن استخدام تعليمات القفز النسب ا عا مات 80×86 فقط عند نقل التحكم.....
- بايتاً بدءاً من التعليمات ا الية.
- (a) ±256 (b) ±128 (c) ±512 (d) ±1024
- 4.20 ب ديد..... العنونة ا وجهة بالإزاحة.
- (a) العنوان ا طلق (b) العنوان القاعدي (c) العنوان الفوري
- 4.21 الإزاحة هي..... يستخدم ساب العنوان الفعال.
- (a) عنوان (b) دليل (c) عدد
- 4.22 يحسب العنوان الفعال من خلال إضافة أو طرح قيمة الإزاحة إ.....
- (a) العنوان القاعدي (b) العنوان الفوري
- (c) العنوان النسب (d) العنوان ا طلق
- 4.23 تستخدم العنونة الدليلية والعنونة القاعدية طرق ساب العنوان الفعال.
- (a) تلفة (b) متطابقة (c) متشابة
- 4.24 تتطلب إزاحة العناوين حساب العنوان الفعال.....
- (a) ع/طرح (b) ضرب/قسمة (c) كل العمليات ا سابية
- 4.25 لا كن دمج العنونة القاعدية والدليلية مع بعضها البعض.
- صح • خطأ
- 4.26 يعت وجود العديد من مسجلات الدليل وحدة ا عا ا ركزية أفضل من أجل معا ا
- (a) صفوفات ا تعددة الأبعاد (b) السلاسل
- (c) القوائم (d) الأشجار
- 4.27 إن عدد مسجلات الدليل ا عا 8086 يساوي
- (a) 1 (b) 2 (c) 3 (d) 4

4.28 إن استخدام العنونة الدليلية أو العنونة القاعدية لا يزيد من زمن تنفيذ التعليمة.

• صح • خطأ

4.29 إن وجود عدد أ ط العنونة ا اسوب سوف.....

(a) يبسط التكوين ا ادي (b) يصعب عمل ا مع (أو ا ترجم)

(c) يبسط كتابة برامج لغة التجميع

4.30 بشكل عام، تساعد أ ط العنونة زيادة كتابة ال امج بلغة التجميع

(a) فعالية (b) مرونة (c) صعوبة