

المصفوفات-نظم معلومات ادارية-المرحلة الثانية- م. زينب صبيح جمعة

المصفوفة : هي عبارة عن مجموعة من المواقع الخزنية في الذاكرة تستخدم وتتصف بما يأتي :

- 1- جميع المواقع تكون من نوع بياني واحد ،حسب صيغة التعريف float ,int ,char.....
- 2- يمكن الوصول عشوائيا (Randomly accessed) الى اي موقع من مواقعها دون الاعتماد على اي موقع في المصفوفة فمقدار الوقت المطلوب للوصول الى اي موقع هو مقدار ثابت .
- 3- مواقع عناصر المصفوفة تبقى ثابتة ولا تتغير اثناء التعامل مع اي من عناصر المصفوفة.
- 4- تمثل المصفوفة في مواقع متعاقبة في الذاكرة .

1- تمثيل المصفوفة الاحادية في الذاكرة :

في لغة C++ تعرف هذه المصفوفة كالاتي :

```
int {or any other type} x [N]
```

Ex:

```
int A[10];
```

```
float A[20];
```

وهذا يعني تعريف هيكل بياني يستوعب مجموعة من العناصر البيانية عددها N مثلا باسم بياني واحد هي X ويستخدم الدليل (index) للوصول الى العناصر البياني المطلوب وتتراوح قيمة الدليل ($1 \leq i \leq n$) وبموجب هذا التعريف يحدد مترجم اللغة (compiler) المنطقة الجزئية لاستيعاب مجموعة العناصر البيانية ويكون المرقم الاول مخصصا للعنصر الاول في المصفوفة وهو ما يطلق عليه عنوان البداية (Base address) (BA) وليكن افتراضا هو (500) اما العنصر الثاني للمصفوفة فيكون عنوانه بعد عنوان البداية مباشرة اي (501) وهكذا بقية العناصر بالتتابع.

$$\text{Location } (x[i]) = \text{Base Address} + (i-1)$$

Ex:

$$\begin{array}{l} A[4] \\ A = [2 \ 3 \ 4 \ 5] \\ A[0]=2 , A[1]=3 , A[3]=4 , A[4]=5 \end{array}$$

عناوين نسبية وليست حقيقية

Real Address عنوان حقيقي



100

2

101

3

102

4

103

5

Memory الذاكرة



الحل:

$i=4$

$$\begin{aligned} \text{Location}(x[4]) &= 500 + (4-1) \\ &= 500 + 3 = 503 \end{aligned}$$

مثال : لتكن x مصفوفة جد العنوان الحقيقي
للعنصر الرابع في المصفوفة علما ان العنوان
الاساسي هو 500

* فعندما يتضمن البرنامج اية اشارة او تعامل مع عناصر المصفوفة في اي ايعاز مثل `cin>>x[i]` و `cout<<x[i]` او غيرهما فان المترجم يعتمد العلاقة المشار اليها لتحديد الموقع المطلوب .

2- تمثيل المصفوفة الثنائية في الذاكرة :

في لغة C++ تعرف هذه المصفوفة كالاتي ،

```
int {or any other type} A[M][N]
```

Ex:

```
Int A[10][10];
```

```
Float A[20][20];
```

وهذا يعني تعريف هيكل بياني اسمه (A) يستوعب مجموعة من العناصر البيانية عددها ($M*N$) ويستخدم دليلين للوصول الى العناصر البياني المطلوب وهما

$1 \leq i \leq M$ لتحديد الصف الذي فيه العنصر

$1 \leq j \leq N$ لتحديد العمود الذي فيه العنصر

فمثلا العنصر $A[3][5]$ حيث $i=3$ و $j=5$

وهذا يعني ان العنصر يقع في الصف الثالث والعمود الخامس .

طريقة تخزين المصفوفة داخل الذاكرة:

Ex:

$A[2][2]$

$$A = \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}$$

$$A[0][0]=2 , A[0][1]=3 , A[1][0]=5 , A[1][1]=6$$

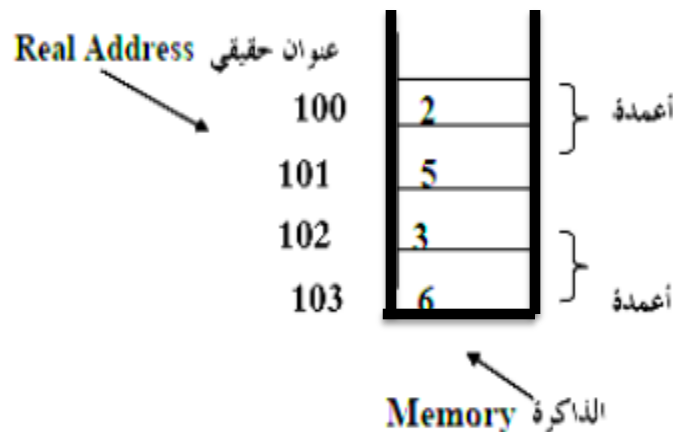
عناوين نسبية وليست حقيقية

التمثيل المنطقي للمصفوفة الثنائية: Logical Structure

$$A = \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0m} \\ a_{10} & a_{11} & \dots & a_{1m} \\ \vdots & \vdots & & \vdots \\ a_{n0} & a_{n1} & \dots & a_{nm} \end{pmatrix} \quad n \times m$$

التمثيل الحقيقي للمصفوفة الثنائية: Physical Structure

(عمود - عمود):



ويعتمد مترجم اللغة (compiler) احدى الطريقتين الاتيتين لتمثيل هذه المصفوفة :

ب) طريقة الاعمدة Column-wise method :

وهذه الطريقة مستخدمة في لغة الفورتران والبيسك.

اذ تؤخذ عناصر العمود الاول ($j=0$) للمصفوفة وتخزن في الذاكرة ثم يتم اخذ عناصر العمود الثاني ليتم خزنها بعد عناصر العمود الاول وهكذا الى نهاية الاعمدة.

وعليه فان احتساب موقع العنصر $A[i][j]$ يكون وفق المعادلة التالية :

$$\text{Location (A [i] [j])} = \text{Base Address} + M \times j + i$$

حيث ان M تمثل عدد الصفوف الكلي للمصفوفة

مثال S : مصفوفة ثنائية من 6 صفوف و 8 اعمدة احسب الموقع الحقيقي لـ $S[5][7]$

Sol:

علما ان العنوان الاساسي هو 300

$$i=5, j=7, M=6, N=8$$

$$\text{Location (s [5] [7])} = 300 + 6 * 7 + 5$$

$$= 300 + 42 + 5 = 347$$

ب- طريقة الصفوف Row-wise method:

وهذه الطريقة مستخدمة في لغة باسكال ،كوبول ، ++C

حيث تؤخذ جميع عناصر الصف الاول ($i=0$) للمصفوفة وتُخزن في الذاكرة ابتداء من موقع البداية (Base Address).
ثم تؤخذ جميع عناصر الصف الثاني ($i=1$) للمصفوفة وتُخزن في الذاكرة بعد اخر عنصر من عناصر الصف الاول وهكذا الى نهاية الصفوف.

ولهذا فان احتساب موقع العنصر $A[i][j]$ يكون وفق المعادلة التالية :

$$\text{Location (A[i] [j])} = \text{Base Address} + N * i + j$$

حيث ان N تمثل عدد الاعمدة الكلي للمصفوفة

Ex: Let int A [5] [7]

Compute the location of the element A [4] [6] ?

When the base Address is 900.

Sol:

$$i=4, j=6, M=5, N=7$$

$$\text{Location (A [4] [6])} = 900 + 7 * 4 + 6$$

$$= 900 + 28 + 6 = 934$$

التخصيص الخزني Storage allocation

1-التخصيص الخزني التسلسلي Sequential storage allocation

يعتبر ابط طرق خزن القوائم الخطية حيث يتم خزن البيانات في مواقع خزنية متتابة (متسلسلة) في ذاكرة الحاسوب .يمكن ان نعرف موقع اي عنصر اذا عرفنا موقع العنصر الاول الذي يمثل عنوان البداية) Base Address (حيث ان مواقع العناصر الاخرى يمكن حسابها من خلال موق البداية .مثلا العنصر) K (سيكون موقعه مباشرة بعد العنصر) k-1 (وهكذا بقية العناصر.

المزايا:

- 1-سهولة التمثيل والتطبيق
- 2-كفاءة عالية في الوصول العشوائي للبيانات
- 3-مناسب جدا عند التعامل مع المكس

المساوي:

1. صعوبة تنفيذ عمليات الاضافة والحذف
2. يتطلب التعريف المسبق وتحديد عدد العناصر المطلوب خزنها

2-التخصيص الخزني الديناميكي Dynamic storage allocation

في طريقة الخزن هذه يتم استخدام رابط link او مؤشر مع كل عنصر يحتوي على عنوان موقع العنصر التالي لذلك لا توجد ضرورة لخزن البيانات في مواقع متعاقبة(متسلسلة) في الذاكرة بل يمكن خزن العنصر (البيانات) في اي موقع في الذاكرة .لذلك كل عنصر يحتوي على جزئين هما:

الجزء الاول : حقل يحتوي على البيانات (Data)

الجزء الثاني : حقل يحتوي على عنوان موقع العنصر التالي (link)

المزايا:

سهولة تنفيذ عمليات الاضافة والحذف لأي عنصر اذ لا يتطلب اكثر من تحديث قيمة حقل المؤشر الذي يطي عنوان الموقع التالي.

المساوي:

يحتاج الي مساحة خزنه اكبر لتمثيل حقل المؤشر اضافة الى البيانات الأساسية

المقارنة بين الخزن التسلسلي والخزن الديناميكي

- 1- **المساحة الخزنية : Amount of storage** ان اسلوب الخزن الديناميكي يحتاج الي مساحة خزنية اكبر لان كل عنصر في الهيكل البياني يحتاج الى مؤشر (اي موقع خزني اضافي) يحتوي على عنوان موقع العنصر التالي وهكذا جميع العناصر.
- 2- **عمليات الاضافة والحذف :** ان هذه العمليات اكثر سهولة في الخزن الديناميكي لانها لا تتطلب سوى تحديث قيمة المؤشر عند الاضافة او الحذف .اما في الخزن التسلسلي فالعملية اكثر تعقيدا لانها تتطلب عملية تزحيف (shifting) للعناصر.
- 3- **الوصول العشوائي للعناصر : Random Access** ان اسلوب الخزن التسلسلي يعتبر افضل في الوصول العشوائي لاي عنصر من عناصر الهيكل البياني مباشرة (k-th element from the start) اما في الخزن الديناميكي فانه يتطلب البدء من اول عنصر ثم العناصر التالية بالتتابع لحين الوصول للعنصر المطلوب.
- 4- **الدمج والفصل : Merge and Split** في الخزن الديناميكي تكون هذه العمليات اسهل تنفيذا وذلك بتغير قيمة المؤشر للعناصر (العقد) في مواقع الدمج او الفصل .اما في الخزن التسلسلي فالعمل اكثر تعقيدا اذ قد يتطلب تزحيف العناصر واعادة تنظيم الخزن Reorganization .