DATA STRUCTURES AND ALGORITHMS

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

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 ولاتتغير اثناءالتعامل 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=<i<=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 العناصربالتتابع.

Location (x[i]) = Base Address + (i-1)

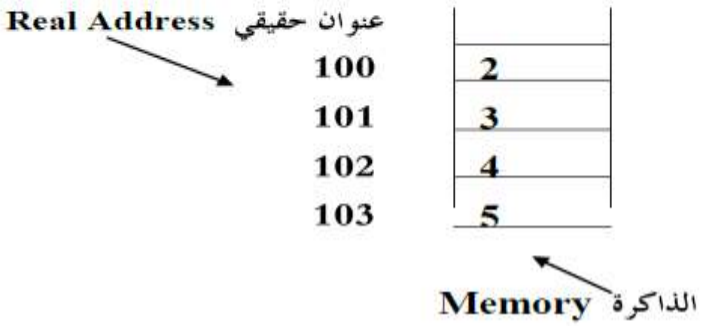
Ex:

A[4]

A=[2 3 4 5]

A[0]=2 , A[1]=3 , A[3]=4 , A[4]=5

عناوين نسبية وليست حقيقية



Ex: Let int x [n] any Array compute the Location of the element x [4] when the bass Address is 500 ?

Solution :

i=4

Location(x [4]) =500+ (4-1)

=500+3=503

* 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=<i<=M لتحديد الصف الذي فيه العنصر

1=<j<=N لتحديد العمود الذي فيه العنصر

فمثلا العنصر A[3][5] حيث i=3 وj=5

وهذا يعني ان العنصر يقع في الصف الثالث والعمود الخامس .

طريقة خزن المصفوفة داخل الذاكرة:

Ex:

A[2][2]

A= [2 3 ; 5 6]

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

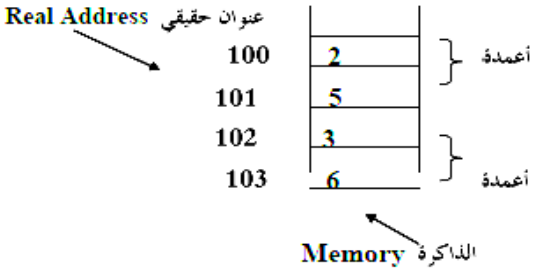
عناوين نسبية وليست حقيقية

التمثيل المنطقي للمصفوفة الثنائية: Logical Structure

$$A= \begin{pmatrix} a00 & a01 & \cdots & a0m \\ a10 & a11 & \cdots & a1m \\ \vdots & \vdots & & \vdots \\ an0 & an1 & \cdots & anm \end{pmatrix} \quad n*m$$

التمثيل الخقيقي للمصفوفة الثنائية: Physical Structure

(عمود – عمود):



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

أ- طريقة الاعمدة Column -wise method :

وهذه الطريقة مستخدمة في لغة الفورتران والبيسك.

اذ توخذ 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

Ex: Let int s [6] [8] any Array, compute the location of the element s [5] [7] when Base Address is (300).

Sol:

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

Location (s [5] [7]) =300+ 6*7 +5

= 300+ 42+5 =347

أ- طريقة الصفوف Row-wise method:

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

حيث توخذ جميع عناصر الصف الاول (i=0) للمصفوفة وتخزن في الذاكرة ابتداء من موقع البداية (Base Address).

ثم توخذ جميع عناصر الصف الثاني (i=1) للمصفوفة وتخزن 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 :

Location (A[i] [j]) = 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

Location (A [4] [6]) = 900+ 7 * 4 + 6

= 900 + 28 + 5 = 934