


Nguyễn Phúc Khải

CHƯƠNG 11: *MẢNG*

Các nội dung:

- Khái niệm
- Khai báo mảng
- Khởi động trị của mảng
- Mảng là đối số của hàm, mảng là biến toàn cục
- Các ứng dụng

KHÁI NIỆM

- Mảng là một biến cấu trúc trong đó có nhiều phần tử cùng kiểu. Mỗi phần tử là một biến thành phần của mảng. Mỗi biến thành phần này là một biến bình thường và có cước số (subscript) để phân biệt giữa phần tử này và phần tử kia. Như vậy, để truy xuất một phần tử của mảng, ta cần biết được cước số của nó.
- Trong bộ nhớ, các phần tử của mảng được cấp phát ô nhớ có địa chỉ liên tiếp nhau.

KHÁI NIỆM

- C cũng cho phép làm việc trên ***mảng một chiều*** (singledimensional array) và ***mảng nhiều chiều*** (multidimensional array).
- Số phần tử trên một chiều được gọi là kích thước của chiều đó.

KHAI BÁO MẢNG

- **Mảng một chiều:**
- Cú pháp khai báo mảng một chiều như sau:

kiểu tên_mảng [kích_thước];

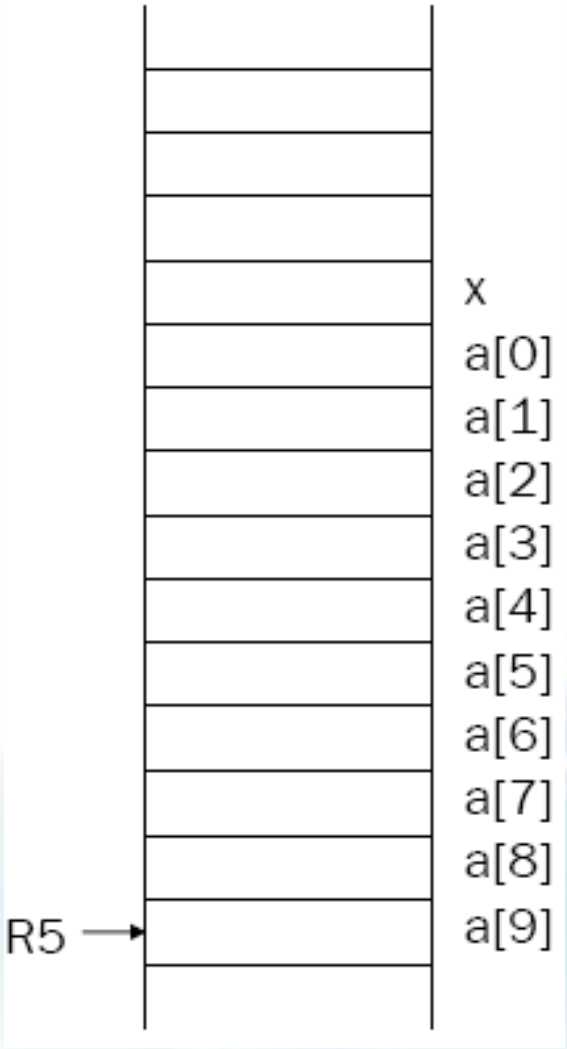
- Với ***kích_thước*** là một hằng số nguyên cụ thể, cho biết số phần tử trong chiều đang xét.
- Trong C, cước số các phần tử của mảng luôn đi ***từ 0 trở đi***, nên mảng một chiều có ***n*** phần tử thì cước số các phần tử của mảng là 0,..., n-1.

KHAI BÁO MẢNG

- Ví dụ: Cho khai báo sau:

  int a[10], x;

- Như vậy mảng a có 10 phần tử **int**, các phần tử đó là a[0], a[1], …, a[9]. Các phần tử này được cấp phát vị trí trong bộ nhớ như hình 12.1 sau.



KHAI BÁO MẢNG

- **Ví dụ :** Viết chương trình nhập một dãy các số nguyên, tìm số lớn nhất trong dãy số đó.

```
#include <stdio.h>
#include <conio.h>
main()
{
    int i, n, max, vtmax;
    int a[100];
    clrscr();
    printf ("Chuong trinh thu mang \n");
```

KHAI BÁO MẢNG

```
printf ("Moi ban nhap so phan tu cua mang: ");
scanf ("%d", &n);
printf ("Moi nhap cac phan tu cua mang:");
for (i = 0; i < n; i++)
      scanf ("%d", &a[i]);
max = a[0];          vtmax = 0;
for (i = 1; i < n; i++)
      if (max < a[i])
            {        max = a[i];
                     vtmax = i;                }
printf ("Phan tu %d co tri lon nhat la %d\n", vtmax, max);
getch() }
```

KHAI BÁO MẢNG

- **Mảng nhiều chiều:**
- Cú pháp khai báo mảng nhiều chiều như sau:

kiểu tên_mảng [kích_thước_chiều1]
[kích_thước_chiều2] [...];

- Khi dịch C báo lỗi: **"Array size too large?"**

KHAI BÁO MẢNG

- **Ví dụ:** Khai báo mảng hai chiều a

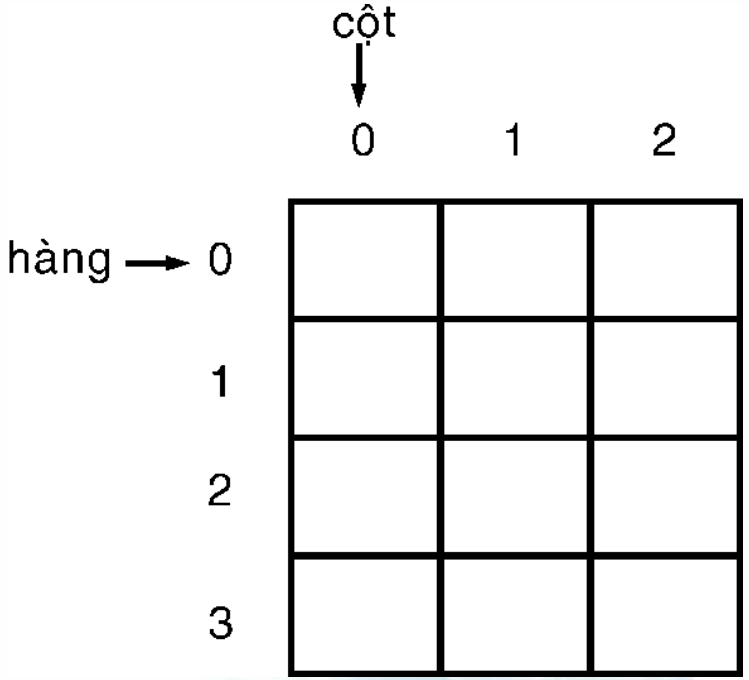
```
int a[4][3];
```

- Như vậy mảng a có 4x3 phần tử **int**, các phần tử đó là:

```
a[0][0] a[0][1] a[0][2]
a[1][0] a[1][1] a[1][2]
a[2][0] a[2][1] a[2][2]
a[3][0] a[3][1] a[3][2]
```

KHAI BÁO MẢNG

- Các phần tử này được sắp trong bộ nhớ theo thứ tự a[0][0], a[0][1], a[0][2], a[1][0], a[1][1], a[1][2], a[2][0], a[2][1], a[2][2],....



KHAI BÁO MẢNG

- **Ví dụ 1:** Viết chương trình tạo và in ra màn hình ma trận đơn vị cấp ***n***

KHAI BÁO MẢNG

```
#include <stdio.h>
#include <conio.h>
#define MAX 20
main()
{   int i, j;
    int a[MAX][MAX];
    int n;
    clrscr();
    printf ("Chuong trinh thu mang \n");
    printf ("Moi ban nhap cap cua ma tran: ");
    scanf ("%d", &n);
```

KHAI BÁO MẢNG

```
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            a[i][j] = (i == j);
    printf ("Ma tran duoc tao la: \n");
        for (i = 0; i < n; i++)
        {       for (j = 0; j < n; j++)
                        printf ("%d", a[i][j]);
            printf("\n");}
            getch ()                }
}
```

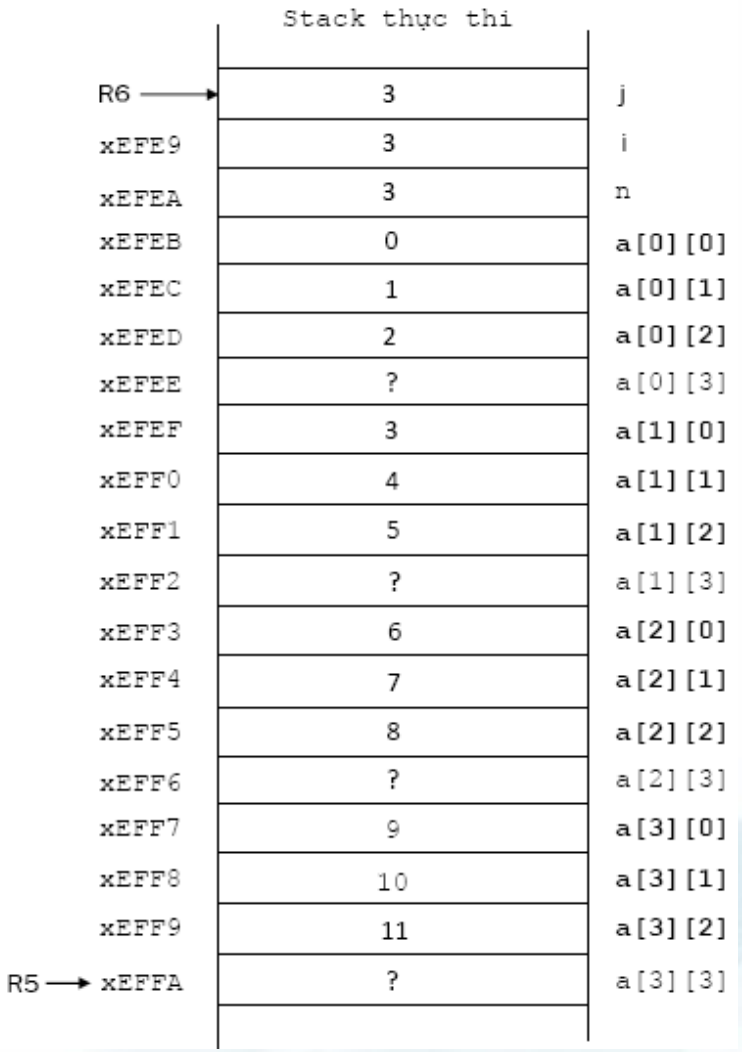
KHAI BÁO MẢNG

- Ví dụ 2:

```
#define MAX 4
        int a[MAX][MAX];
        int n = 3;/* cấp thực sự cần làm việc của ma trận  */
        int i, j;     /* biến là chỉ số mảng  */
    /* Nhập trị cho mảng*/
  for (i = 0; i < n; i++)
            for (j = 0; j < n; j++)   scanf ("%d", &a[i][j]);
```

KHAI BÁO MẢNG

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]
a[3][0]	a[3][1]	a[3][2]	a[3][3]



KHAI BÁO MẢNG

- **Ví dụ :**

```
int a[10];
```

- mà ta lại thực hiện lệnh

```
for (i = 0; i<= 10; i++) a[i] = i;
```

- thì trong thực tế không có phần tử a[10], nhưng ***chương trình không báo lỗi***, việc gán cũng được thực hiện, và ô nhớ kế tiếp phần tử a[9] được gán trị.

KHAI BÁO MẢNG

- C không có sự phân biệt giữa một biến chuỗi và một mảng các ký tự. Cả hai trường hợp đều được khai báo:

char tên [chiều_dài];

- Hàm **gets()** cho phép nhập một chuỗi có tên để trong đối số hàm này.
- Hàm **puts()** cho phép xuất một chuỗi có tên để trong đối số hàm này ra màn hình.
- Cả hai gets() và puts() đều có prototype nằm trong file *stdio.h*.

KHAI BÁO MẢNG

- **Bài tập:** Viết chương trình truy xuất chuỗi dùng hàm chuẩn của C.

KHỞI ĐỘNG TRỊ CỦA MẢNG

- Khi khai báo mảng là biến toàn cục hoặc tĩnh thì mảng có thể được khởi động trị bằng các giá trị hằng.
- **Ví dụ :**

```
int a[5] = {1, 3, 5, 7, 9};
int b[10] = {1, 2, 3, 4, 5};
```

- Nếu số trị ít hơn số phần tử mảng thì các phần tử còn lại không được khởi động trị, có nghĩa các phần tử này có trị là 0.

KHỞI ĐỘNG TRỊ CỦA MẢNG

- Ví dụ:

```
double a[] = {1.23, –5.67, 9.87, 1.34};
char s[30] = “I go to school \n”;
char ch[] = “Hello, World!”;
```

KHỞI ĐỘNG TRỊ CỦA MẢNG

- Ví dụ: Khai báo mảng 2 chiều:

```
int a[][3] = {
                { 11, 12, 13},
                { 21, 22, 23},
                { 31, 32, 33}
            };
```

Với khai báo này, mảng a sẽ có 9 phần tử trong 3 hàng

MẢNG LÀ ĐỐI SỐ CỦA HÀM
MẢNG LÀ BIẾN TOÀN CỤC

- Khi khai báo đối số của hàm là mảng, kích thước của chiều đầu tiên của mảng không cần xác định cụ thể. Tuy nhiên từ chiều thứ hai trở đi, kích thước mảng phải xác định.
- Tên mảng chính là địa chỉ của mảng, nên ***việc truyền tên mảng*** cho hàm chính là ***truyền địa chỉ thực của mảng nên mọi thay đổi trên mảng trong hàm cũng chính là thay đổi trên mảng thật*** (truyền theo kiểu tham số biến).

MẢNG LÀ ĐỐI SỐ CỦA HÀM
MẢNG LÀ BIẾN TOÀN CỤC

```
void select_sort (int a[ ], int n)
        {       int i, j;    int max, vtmax; int tam;
                for (i = 0; i < n-1; i++)
                        {       max = a[i]; vtmax = i;
                                for (j = i + 1; j < n; j++)
                                        if (max < a[j])
                                                {       max = a[j];
                                                        vtmax = j;   }
                                if (vtmax ! = i)
                                {       tam = a[i];
                                        a[i] = max;
                                        a[vtmax] = tam;   }
                        }
        }
```

CÁC ỨNG DỤNG

- Sắp xếp mảng:
  - Select sort
  - Bubble sort
  - Quick sort
- Các ứng dụng khác

Kết thúc chương 11