行指针与列指针

 (2013-05-09 18:14:24)

[[](javascript:;)转载▼](javascript:;)

|  |  |
| --- | --- |
| 标签：  教学 | 分类： [教学内容](http://blog.sina.com.cn/s/articlelist_1934585941_2_1.html) |

行指针和列指针是我们在学习指针处理二维数组时遇到的知识难点。在这里我结合资料为大家做一些分析。

首先数组的访问有三种形式：1）下标法；2）首地址法；3）指针变量法。这里我们先分析首地址法。

假设有一个二维数组a[3][4]，a就是首地址，他是一个指针是一个常量，指向第一个元素的地址，也就是指向第一行的首地址，是指首行一整行，并不是指某个具体元素。那么我们称之为“行指针”。同理：a+0,a+1,a+2,都是行指针如下表所示。

**结论：**

|  |  |  |
| --- | --- | --- |
| **表示形式** | **含义** | **指针类型** |
| **a或者a+0** | **指向第0行** | **行指针** |
| **a+1** | **指向第1行** | **行指针** |
| **a+2** | **指向第2行** | **行指针** |

但我们发现光有行指针是不能对列元素进行访问的。所以我们再来看a[0]，他也就是这个数组的首地址，也就是第一个元素的地址，只不过指针方向是指向列的方向。也就是a[0]+0。a[0]和a[0]+0都是指具体的元素，那么我们称之为“列指针”，如下表所示。

**结论：（第0行视为一维数组）**

|  |  |  |
| --- | --- | --- |
| **表示形式** | **含义** | **指针类型** |
| **a[0]** | **是一维数组的名称，也是它的首地址，而且是第1个元素的地址（a[0]+0）** | **列指针** |
| **a[0]+1** | **第0行，第2个元素的地址** | **列指针** |
| **a[0]+2** | **第0行，第3个元素的地址** | **列指针** |

**那么两个概念之间的具体转换是：**

**\*行指针----列指针**

**&列指针----行指针**

**对应关系如下表所示**

**根据以上转换公式:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **行指针** | **转换成：列指针** | **列指针等价表示** | **内容** | **内容等价表示** | **含义** |
| **a或a+0** | **\*a** | **a[0]** | **\*a[0]** | **\*(\*a)** | **a[0][0]** |
| **a+1** | **\*(a+1)** | **a[1]** | **\*a[1]** | **\*(\*(a+1))** | **a[1][0]** |
| **a+2** | **\*(a+2)** | **a[2]** | **\*a[2]** | **\*(\*(a+2))** | **a[2][0]** |

**对于元素a[1][2],其地址用列指针表示为a[1]+2，等价表示为\*(a+1)+2，那么内容是\*(\*(a+1)+2)；**

|  |  |  |  |
| --- | --- | --- | --- |
| **列指针** | **行指针** | **等价表示** | **含义** |
| **a[0]** | **&a[0]** | **&a或&(a+0)** | **第0行** |
| **a[1]** | **&a[1]** | **&(a+1)** | **第1行** |
| **a[2]** | **&a[2]** | **&(a+2)** | **第2行** |

**示例1：用列指针输出二维数组。**

**#include**

**void main()**

**{**

**int a[3][4]={1,3,5,7,9,11,13,15,17,19,21,23};**

**int \*p= a[0];  //** **列指针的定义法**

**for(; p < a[0] + 12; p++)**

**{**

**printf("%d ",\*p);**

**}**

**return;**

**}**

**示例2：用行指针输出整个二维数组。**

**#include**

**void main()**

**{**

**int a[3][4]={1,3,5,7,9,11,13,15,17,19,21,23};**

**int (\*p)[4]= &a[0]; // 行指针定义法或者int (\*p)[4]= a;**

**int i, j;**

**for(i = 0; i < 3; i++)**

**for(j = 0; j < 4; j++)**

**{**

**printf("%d ",\*(\*(p + i) + j));**

**}**

**return;**

**}**