

二维数组的指针

杨振平

二维数组的指针

设二维数组: `int a[3][4], i, j;`

1. 二维数组a中i行j列元素的地址- `&a[i][j]`

二维数组在内存中映射为一个一维数组, 因此可以通过指向元素的指针快速访问二维数组中的每个元素。

如: 利用指向数组元素类型的指针变量p, 寻找a数组中元素的最大值
假设:

```
int *p, max=a[0][0]; //max为最大值
for(p=&a[0][0];p<&a[0][0]+12;p++)
    if(*p>max)
        max=*p;
cout<<"max="<<max<<endl;
```

二维数组的指针(续)

2. 二维数组的行地址

int a[3][4]; 由3行元素组成, 即a – a[0],a[1],a[2]

而每行又由4个类型相同的元素组成, 分别对应一个一维数组。

a[0]– a[0][0],a[0][1],a[0][2],a[0][3]

a[1]– a[1][0],a[1][1],a[1][2],a[1][3]

a[2]– a[2][0],a[2][1],a[2][2],a[2][3]

其中:

a为行元素数组的名字,即a代表&a[0],即0行的地址

a+1代表&a[1],即1行的地址

a+2代表&a[2],即2行的地址。

二维数组元素的指针法表示

由于 $a[0]$ 是由 $a[0][0]$, $a[0][1]$, $a[0][2]$ 和 $a[0][3]$ 四个元素构成的一维数组

因此, $a[0]$ 代表 $\&a[0][0]$,即0行的首元素的地址

这样 $a[0]+1$ 代表首元素的下一个元素的地址即 $\&a[0][1]$ 而 $a[0]+j$ 代表 $\&a[0][j]$

同理, $a[1]$ 代表 $\&a[1][0]$,即1行的首元素的地址 而 $a[1]+j$ 代表 $\&a[1][j]$

$a[i]$ 代表 $\&a[i][0]$,即*i*行的首元素的地址 而 $a[i]+j$ 代表 $\&a[i][j]$

由此我们得到: $*(a[i]+j)$ 等价于 $a[i][j]$

又由于 $a[i]$ 等价于 $*(a+i)$, 因此, $*(a[i]+j)$ 也等价于 $*(*(a+i)+j)$

即: $*(*(a+i)+j)$ 与 $a[i][j]$ 等价

我们将 $*(*(a+i)+j)$ 称为二维数组元素 $a[i][j]$ 的指针法表示。

其中, a 为首行地址, $a+i$ 为*i*行的行地址, 而 $*(a+i)$ 为*a*的*i*行0列元素的地址, 而 $*(a+i)+j$ 为*a*的*i*行*j*列元素的地址。