多维数组的地址

设有整型二维数组a[3][4]如下：

0 1 2 3

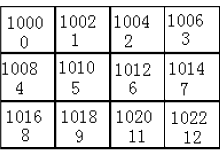
4 5 6 7

8 9 10 11

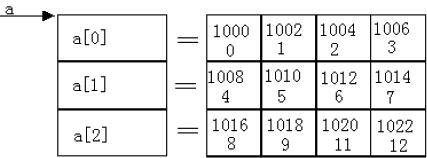
它的定义为：

int a[3][4]={{0,1,2,3},{4,5,6,7},{8,9,10,11}}

设数组a的首地址为1000，各下标变量的首地址及其值如图所示。

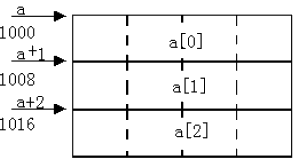


前面介绍过，C语言允许把一个二维数组分解为多个一维数组来处理。因此数组a可分解为三个一维数组，即a[0]、a[1]、a[2]。每一个一维数组又含有四个元素。



例如a[0]数组，含有a[0][0]，a[0][1]，a[0][2]，a[0][3]四个元素。

数组及数组元素的地址表示如下：从二维数组的角度来看，a是二维数组名，a代表整个二维数组的首地址，也是二维数组0行的首地址，等于1000。a+1代表第一行的首地址，等于1008。如图：



a[0]是第一个一维数组的数组名和首地址，因此也为1000。\*(a+0)或\*a是与a[0]等效的， 它表示一维数组a[0]0 号元素的首地址，也为1000。&a[0][0]是二维数组a的0行0列元素首地址，同样是1000。因此，a，a[0]，\*(a+0)，\*a，&a[0][0]是相等的。

同理，a+1是二维数组1行的首地址，等于1008。a[1]是第二个一维数组的数组名和首地址，因此也为1008。&a[1][0]是二维数组a的1行0列元素地址，也是1008。因此a+1,a[1],\*(a+1),&a[1][0]是等同的。

由此可得出：a+i，a[i]，\*(a+i)，&a[i][0]是等同的。

此外，&a[i]和a[i]也是等同的。因为在二维数组中不能把&a[i]理解为元素a[i]的地址，不存在元素a[i]。C语言规定，它是一种地址计算方法，表示数组a第i行首地址。由此，我们得出：a[i]，&a[i]，\*(a+i)和a+i也都是等同的。

另外，a[0]也可以看成是a[0]+0，是一维数组a[0]的0号元素的首地址，而a[0]+1则是a[0]的1号元素首地址，由此可得出a[i]+j则是一维数组a[i]的j号元素首地址，它等于&a[i][j]。

由a[i]=\*(a+i)得a[i]+j=\*(a+i)+j。由于\*(a+i)+j是二维数组a的i行j列元素的首地址，所以，该元素的值等于\*(\*(a+i)+j)。

【例10-22】

int main(){

int a[3][4]={0,1,2,3,4,5,6,7,8,9,10,11};

printf("%d,",a);

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

printf("%d,",a[0]);

printf("%d,",&a[0]);

printf("%d\n",&a[0][0]);

printf("%d,",a+1);

printf("%d,",\*(a+1));

printf("%d,",a[1]);

printf("%d,",&a[1]);

printf("%d\n",&a[1][0]);

printf("%d,",a+2);

printf("%d,",\*(a+2));

printf("%d,",a[2]);

printf("%d,",&a[2]);

printf("%d\n",&a[2][0]);

printf("%d,",a[1]+1);

printf("%d\n",\*(a+1)+1);

printf("%d,%d\n",\*(a[1]+1),\*(\*(a+1)+1));

}

指向多维数组的指针变量

把二维数组a分解为一维数组a[0]、a[1]、a[2]之后，设p为指向二维数组的指针变量。可定义为：

int (\*p)[4]

它表示p是一个指针变量，它指向包含4个元素的一维数组。若指向第一个一维数组a[0]，其值等于a,a[0]，或&a[0][0]等。而p+i则指向一维数组a[i]。从前面的分析可得出\*(p+i)+j是二维数组i行j 列的元素的地址，而\*(\*(p+i)+j)则是i行j列元素的值。

二维数组指针变量说明的一般形式为：

类型说明符 (\*指针变量名)[长度]

其中“类型说明符”为所指数组的数据类型。“\*”表示其后的变量是指针类型。“长度”表示二维数组分解为多个一维数组时，一维数组的长度，也就是二维数组的列数。应注意“（\*指针变量名）”两边的括号不可少，如缺少括号则表示是指针数组（本章后面介绍），意义就完全不同了。

【例10-23】

main(){

int a[3][4]={0,1,2,3,4,5,6,7,8,9,10,11};

int(\*p)[4];

int i,j;

p=a;

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

for(j=0;j<4;j++) printf("%2d ",\*(\*(p+i)+j));

printf("\n");

}

}