

全国计算机等级考试二级:c\_c 深入探讨数组内存模型 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/181/2021\\_2022\\_\\_E5\\_85\\_A8\\_E5\\_9B\\_BD\\_E8\\_AE\\_A1\\_E7\\_c97\\_181122.htm](https://www.100test.com/kao_ti2020/181/2021_2022__E5_85_A8_E5_9B_BD_E8_AE_A1_E7_c97_181122.htm) 计算机等级考试训练软件《百宝箱》对于一个好学的人来书，怎么做不重要，关键是为什么要这样做，只有深入到这一步了，才敢说‘阿，我懂了’，于是，以后碰到类似的问题，就可以从我们知道的原理出发，悠然自得而又满怀信心地推导出正确的结果。然而，好学的人往往注定了困惑于一时的不解，并且，付出过多的时间去试图解决这种迷惑不解，其中的辛苦是可想而知的，因为，我就是这样走过来的。所以，一直都想写一个关于指针，数组，地址，左右值等等这些在语言的学习里面最让人困惑不解的问题。怎奈时间有限，一直没有能完成这样的想法，今天一样是太忙，可能不能完整的把我想说的写出来了，那么，就走个捷径，抛开c/C的语义，从系统底层的原理来就事论事的解决一下网友的问题，本人水平不高，全当抛砖引玉。[注意，下面的讲解全部以32位系统为例，也就是说，一个int占用4 bytes] 问题：1. 多维数组传值我知道2维数组传值func(type [][]SIZE),但是3维的怎么办？3维以上？若使用类似以上方法，函数中得到的是有const修饰的，也就是说不能更改数组中的值。若用指针怎么写，我知道二维func(type\*\* array),但数组不能直接定义成array[[[]],要用循环，麻烦。3维以上怎么办？数组怎么定义？2. 有什么函数可以高效复制一个数组出来？不然只有再去循环了首先讲讲c/c对数组变量的解释，我们知道语言对变量的解释虽然都是对某一段内存的‘标识’，但是，对于不同的类型的变量，编译

器对他的使用和解释都是有区别的。比如：`void main(){int a=0.int array[10].a=3.//okarray[0]=4.//okarray={1,2,4}.//errorprintf("%d,%d,%d",a,array[0],array).//okreturn.}`对于上面的程序段，`a`代表了系统中一个4byte的内存区域，编译时候用`a`来代表内存的值[也就是所谓右值]，`array`来代表一段（4\*10）byte内存区域的值，而具体到`array`这个变量的身上，编译器解释他的时候，把它看成是这段内存的首地址。所以上面的程序段中的那个printf打印出来的因该是`a`的值，`array`第一个元素的值,`array`的首地址。基于以上的一些事实和理论，`int *p=array.int **p1=amp.array.//`一个指向了‘指向数组的指针’得指针。语义上的不同，对于编译器来解释这个变量的行为的时候非常重要，但是对于系统的内部来说，基本的内存的结构一样的，我们如果能抓住这点的话，那么，指针对我来说将是透明的。下面举个例子：`int array[10].int array2d[3][10].`这两个数组的区别和联系分别是什么，如果你能很清楚地认识到那么，你已经有很好的功力了。首先，不同点，最表面的语义上的不同就是一个是1维，一个是二维，`for(int i=0.i{for(int j=0.j{array2d[j][i]=array[i].}}`上面的程序将有3行数据的`array2d`数组每行都设成和`array`相同的值，从这一层来看，我们更能清晰地感觉到行和列的存在[也就是意识到了维的存在。但是在内存内部的实现又是什么样的情况呢[注意，现在来说相同点啦。我们知道，只要是数组，那么系统将会分配给连续的空间，不管是几维的数组对于`int array[10]`，系统分配了4\*10 bytes的连续空间。对于`int array2d[3][10]`,系统分配了4\*(10\*3)bytes的连续空间,注意，两者的内存模型完全一样

，只是可用的长度不同而已，只要找到这两个连续的空间的头位置，就可以根据偏移来访问任何一个元素，不管是几维的数组，并且，数组名永远指代连续空间的头地址，于是，我们定义 $\text{int}^* p = \text{array}$ . $*p == \text{array}[0]$ . $*(p + 1) == \text{array}[1]$ . $*(p + 9) == \text{array}[9]$ ..... $\text{array}[x] = *(p + x)$ ./////////// 同样， $\text{int}^* p = \text{array2d}$ . $*p == \text{array}[0][0]$ . $*(p + 1) == \text{array}[0][1]$ . $*(p + 9) == \text{array}[0][9]$ . $*(p + 15) == \text{array}[1][5]$ ..... $\text{array}[x][y] = *(p + (10 * x) + y)$ .推广到3维，4维，道理都是一样的,比如， $\text{int array}[A][B][C][D]$ . $\text{int}^* p = \text{array}$ . $\text{array}[a][b][c][d] = *(p + a * B * C * D + b * C * D + c * D + d)$ .后面的部分 ‘ $a * B * C * D + b * C * D + c * D + d$ ’ 称作偏移量 其实，编译器生成的代码中对于数组元素的访问，就是通过上面的等式完成的，到现在，你可能也感觉到了语义层面上的区别，以及他们的重要性了。清晰地说明如下： $\text{int array}[2][3]$ .和 $\text{int array1}[3][2]$ . 对系统来说，都是申请了一个 $2 * 3 * 4 = 24$ bytes的连续空间，但是由于定义规范的不同导致了，内存偏移计算不一样。考察 $\text{array}[1][1]$ 和 $\text{array1}[1][1]$ 的偏移， $\text{array}[1][1]$ 的偏移是 $1 * 3 + 1 = 4$ .(4个int型长的偏移，也就是 $4 * 4 = 16$ byte的偏移) $\text{array1}[1][1]$ 的偏移是 $1 * 2 + 1 = 3$ .所以，数组的定义在内存访问的层面上来讲，就是决定了元素偏移的计算方法。问题解答：1. 多维数组传值 对于数组这样的东西，系统并不是值传递的，[如果说值传递的话，那么也是数组地址的值传递]，也就是说，并没有建立一个完全一样的拷贝数组，而是将数组的地址作为一个值压入栈，供函数使用。所以，你定义的时候，只要定一个指针传入，然后后面给出维数和各维大小就可以了。比如要出入一个4维数组，你可以定义成这个样子 $\text{void fun}(\text{int}^* p, \text{int} d1, \text{int} d2, \text{int} d3, \text{int} d4)$ .然后，在

函数内部，直接使用偏移来访问各个元素，于是，当你要传入int array[A][B][C][D].时候，你可以这样调用fun(array,A,B,C,D). 根据上面的方法，你还可以用不定参数函数的方法，或者用维数数组和数组长的方法来定义出适合任意维数组的寒暑，这个问题留给读者自己去完成了:-).2. 有什么函数可以高效复制一个数组出来？不然只有再去循环了。其实这个问题到这里不言自明了，仅仅举个例子int array1[A][B][C][D].int array2[A][B][C][D]. 拷贝array1内容到array2中去，由于下面使用到了系统相关的底层操作的缘故，假定在win32系统下，太简单了，一句话搞定memmove(array2,array1,A\*B\*C\*D).最后问一个问题，为什么用，memmove不用memcpy?水平有限加之时间仓促，不妥之处在所难免，请各位多指教！100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)