

数组指针和指针数组的区别，C语言数组指针和指针数组区别详解

对指针数组和数组指针的概念，相信很多C程序员都会混淆。下面通过两个简单的语句来分析一下二者之间的区别，示例代码如下所示：

```
1. int *p1[5];  
2. int (*p2)[5];
```

首先，对于语句 “int*p1[5]” ，因为 “[]” 的优先级要比 “*” 要高，所以 p1 先与 “[]” 结合，构成一个数组的定义，数组名为 p1，而 “int*” 修饰的是数组的内容，即数组的每个元素。也就是说，该数组包含 5 个指向 int 类型数据的指针，如图 1 所示，因此，它是一个指针数组。

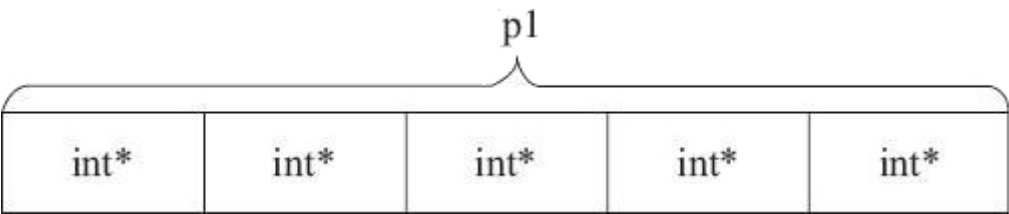


图 1

其次，对于语句 “int(*p2)[5]” ， “()” 的优先级比 “[]” 高，“*” 号和 p2 构成一个指针的定义，指针变量名为 p2，而 int 修饰的是数组的内容，即数组的每个元素。也就是说，p2 是一个指针，它指向一个包含 5 个 int 类型数据的数组，如图 2 所示。很显然，它是一个数组指针，数组在这里并没有名字，是个匿名数组。

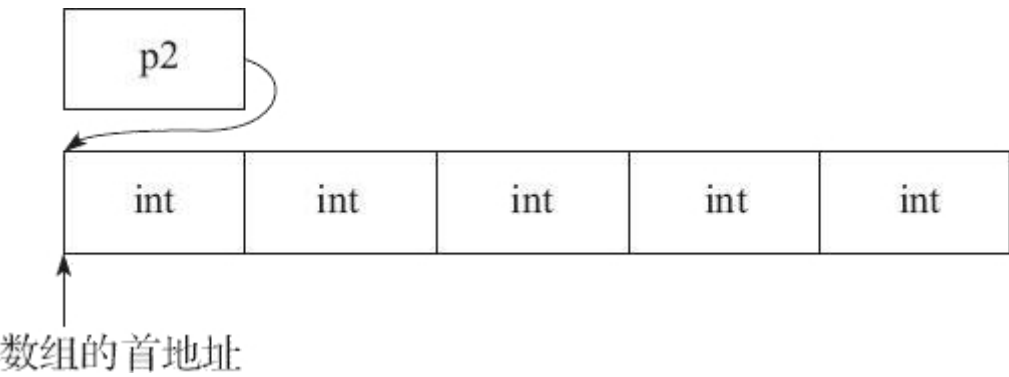


图 2

由此可见，对指针数组来说，首先它是一个数组，数组的元素都是指针，也就是说该数组存储的是指针，数组占多少个字节由数组本身决定；而对数组指针来说，首先它是一个指针，它指向一个数组，也就是说它是指向数组的指针，在 32 位系统下永远占 4 字节，至于它指向的数组占多少字节，这个不能够确定，要看具体情况。

了解指针数组和数组指针二者之间的区别之后，继续来看下面的示例代码：

```
1. int arr[5]={1, 2, 3, 4, 5};  
2. int (*p1)[5] = &arr;  
3. /*下面是错误的*/  
4. int (*p2)[5] = arr;
```

不难看出，在上面的示例代码中，&arr 是指整个数组的首地址，而 arr 是指数组首元素的首地址，虽然所表示的意义不同，但二者之间的值却是相同的。那么问题出来了，既然值是相同的，为什么语句 “int(*p1)[5]=&arr” 是正确的，而语句 “int(*p2)[5]=arr” 却在有些编译器下运行时会提示错误信息呢（如在 Microsoft Visual Studio 2010 中提示的错误信息为 “a value of type”int*”cannot be used to initialize an entity of type”int(*)[5]” ）？

其实原因很简单，在 C 语言中，赋值符号 “=” 号两边的数据类型必须是相同的，如果不同，则需要显示或隐式类型转换。在这里，p1 和 p2 都是数组指针，指向的是整个数组。p1 这个定义的 “=” 号两边的数据类型完全一致，而 p2 这个定义的 “=” 号两边的数据类型就不一致了（左边的类型是指向整个数组的指针，而右边的数据类型是指向单个字符的指针），因此会提示错误信息。