**C指针（三年没下雨版）**

**目录索引**

**一．指针是什么**

**二．指针的作用**

**三．闲话少叙**

**1．指针基本形式**

**2.指针和数组**

**3.指针和指向指针的指针（二维指针）**

**4.函数指针和指针函数**

**四.文终**

**五.本文参考**

**一．指针是什么**

指针和函数是C语言灵魂。

什么是指针，是[内存地址](https://baike.baidu.com/item/%E5%86%85%E5%AD%98%E5%9C%B0%E5%9D%80/7354236)，指针变量是用来存放内存地址的变量，在同一CPU构架下，不同类型的指针变量所占用的存储单元长度是相同的，而存放数据的变量因数据的类型不同，所占用的[存储空间](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E7%A9%BA%E9%97%B4/10657950)长度也不同。有了指针以后，不仅可以对数据本身，也可以对存储数据的变量地址进行操作。指针描述了数据在内存中的位置，标示了一个占据存储空间的实体，在这一段空间起始位置的相对距离值。在 C/C++语言中，指针一般被认为是指针变量，指针变量的内容存储的是其指向的对象的首地址，指向的对象可以是变量（指针变量也是变量），数组，函数等占据存储空间的实体。

很干涩，基本没接触过读起来会无感，下面我们细细来分析一下，相信会有很大收获。

**二．指针的作用**

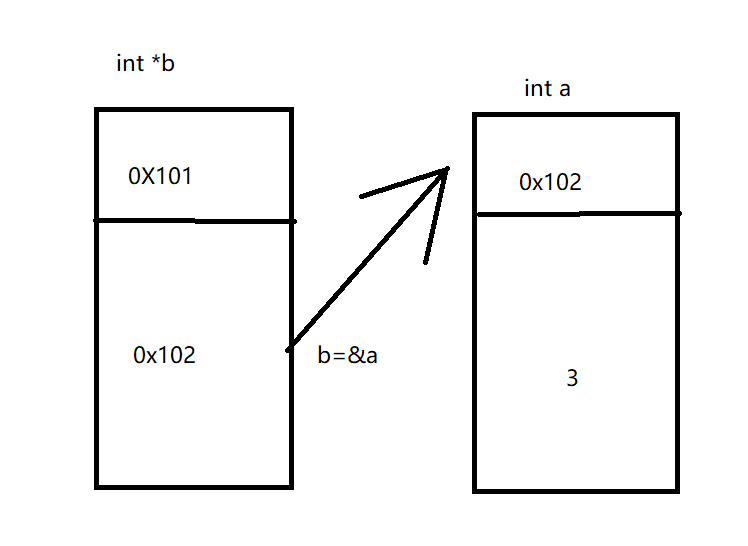
复杂的数据结构，系统地动态分配内存、消息机制、任务调度、灵活矩阵定时都离不开指针，MCU应用中，大一点的项目，需要功能模块化，硬件驱动层和应用层分别独立运行。代码移植性非常强，是离不开指针的。

**三．闲话少叙**

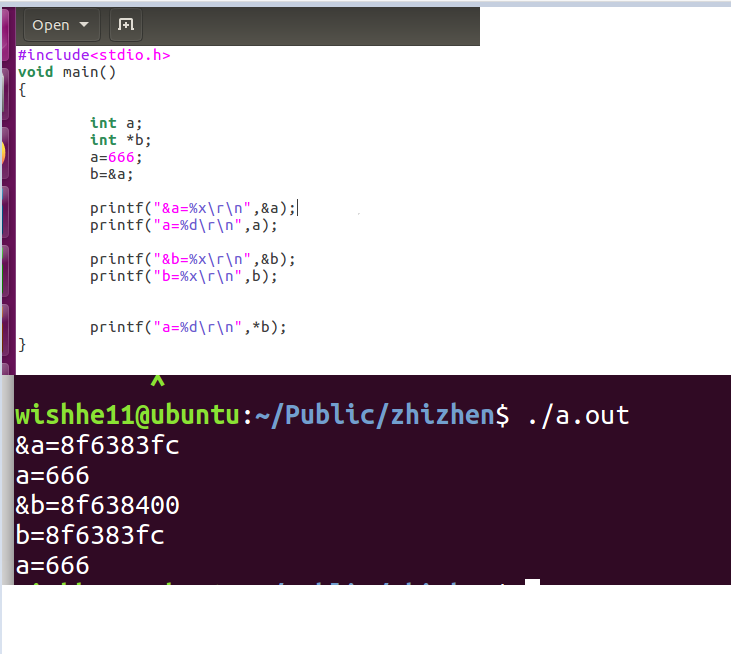
**1．指针基本形式**

先来看一张图片，这里地址不是具体的就是一个例子。最简单的指针就是下面的过程，我们所看到的数据都有他自己的地址和值，指针也拥有自己的地址，只不过他的数据部分存放的是指针。

一般情况下我们使用一个指针的流程是给他定义，int\*b，给他指向的地址空间，b=&a，这里我们需要提前将指针指向的空间开辟出来，可以是如下情况先有int a存在，那么它自然会有一个分配好的地址，来让指针指，如果想实现键盘输入的话，我们需要用malloc来动态开辟输入空间。**反正，无论如何，指针不能是一个无指向的指针**。现在我们先来了解数据在程序里面就初始化好的情况。



上一段代码和执行结果，结果很明显。



**2.指针和数组**

数组和指针关系是极为密切的，很多课本将二者割裂开来，这是对理解是有很大阻碍的。

在开始上操作之前先明确一个指针数组与数组指针的区别。这个基本面试不说一半，也有1/4会问。

指针数组：首先它是一个数组，数组的元素都是指针，数组占多少个字节由数组本身的大小决定，每一个元素都是一个指针，在32 位系统下任何类型的指针永远是占4 个字节。它是“储存指针的数组”的简称。

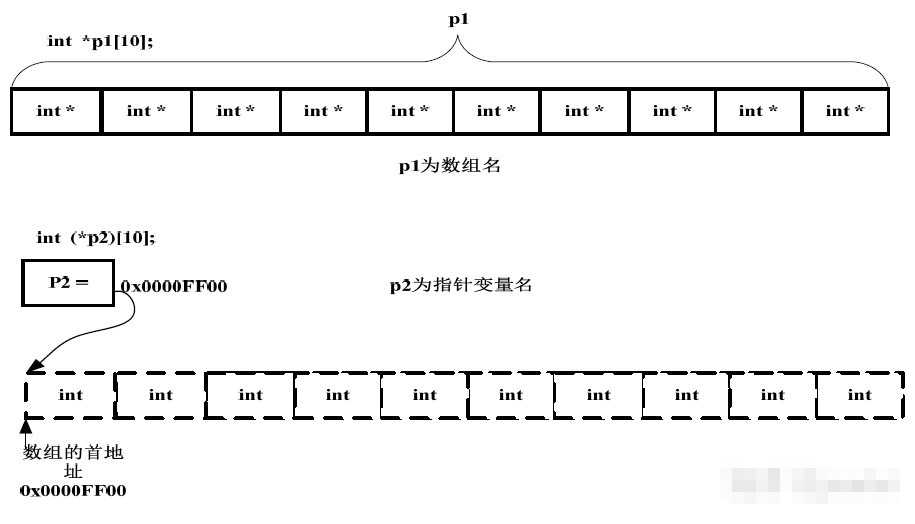
数组指针：首先它是一个指针，它指向一个数组。在32 位系统下任何类型的指针永远是占4 个字节，至于它指向的数组占多少字节，不知道，具体要看数组大小。它是“指向数组的指针”的简称。

做个选择题，哪个是指针数组呢**A)int \*p1[10]; B)int (\*p2)[10];**

这里需要明白一个符号之间的优先级问题。“[]”的优先级比“\*”要高。p1 先与

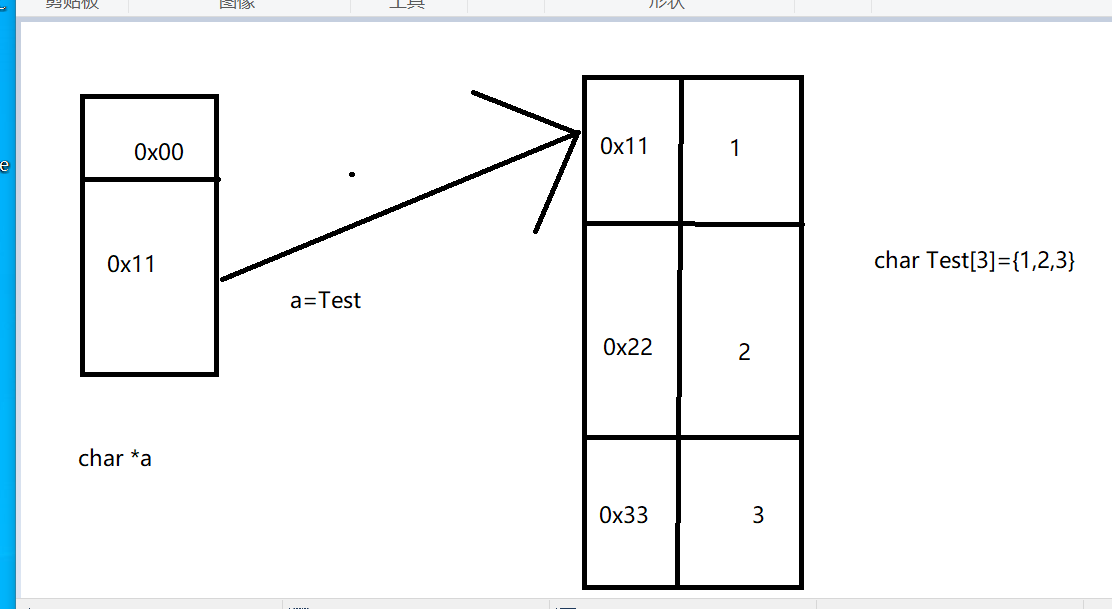
“[]”结合，构成一个数组的定义，数组名为p1，int \*修饰的是数组的内容，即数组的每个元素。那现在我们清楚，这是一个数组，其包含10 个指向int 类型数据的指针，即指针数组。至于p2 就更好理解了，在这里“（）”的优先级比“[]”高，“\*”号和p2 构成一个指针的定义，指针变量名为p2，int 修饰的是数组的内容，即数组的每个元素。数组在这里并没有名字，

是个匿名数组。那现在我们清楚p2 是一个指针，它指向一个包含10 个int 类型数据的数组，即数组指针。

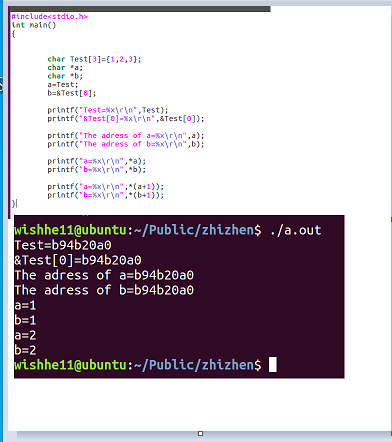


继续上图。

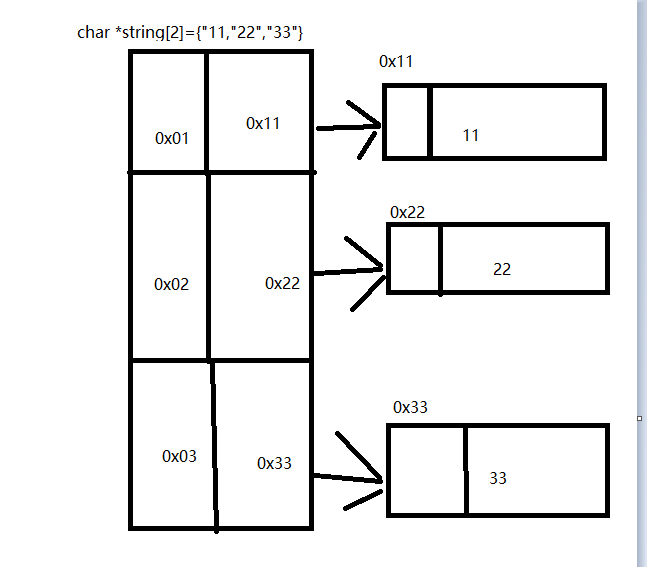
然后开始正题，从数组指针开始。



然后是操作，一样的定义char Test[3]={}, char \*a, a=Test。然后取了一些地址和一些值。总之无论何时一定要记得给指针一个明确的指向。



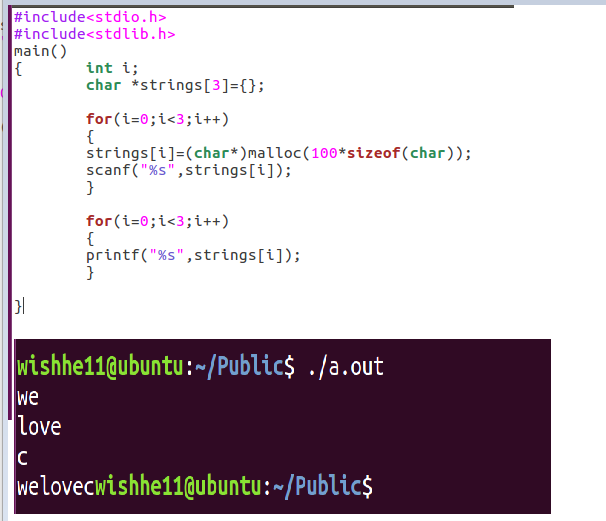
继续，我们看一下什么是指针数组。这里的地址都是一个假设替代，真实的地址还是要去看具体运行的结果。



然后是程序和运行结果。同样的定义类型，确定指向。值得一提的是，假如这个情况下，想从键盘中输入字符串中的内容，我们如果将字符串赋值为空会发生指针无指向，数据越界的情况。



那么为了防止越界，指针无定向的情况出现，我们需要malloc来开辟一些地址给我们键盘输入的数据。下面来操作演示一下程序和操作结果。



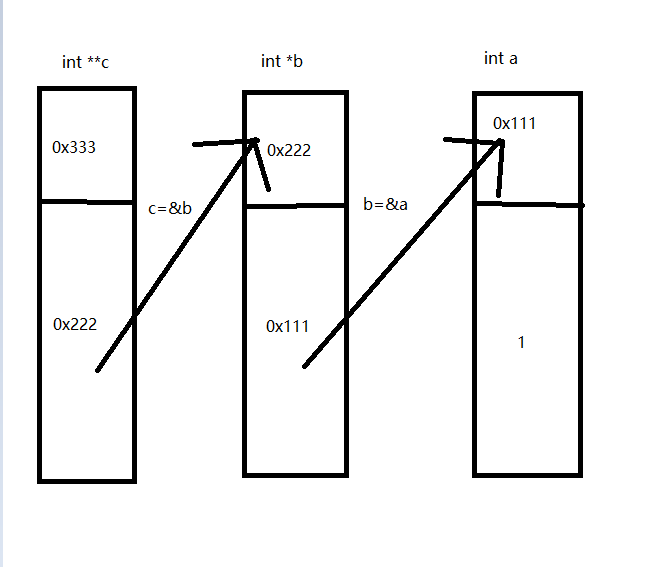
这些数组和指针的操作其实还有很多很多，但是都是大同小异，只有掌握核心原则，以一推三没问题，包括还有二维指针的应用。

**3.指针和指向指针的指针（二维指针）**

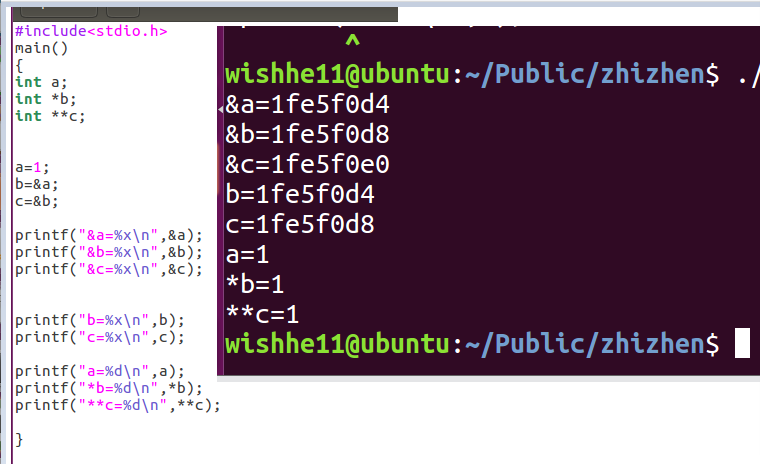
编译器在编译的时，会为指针变量和双重指针变量分配一个存储空间。这里是指针最让人迷惑的原因就是他本身放的是地址，然后本身也是拥有一个地址，不管几维指针只要记住，他本身有地址，他放的地址，就会有一个分析的准则线。

下面来一个具体的程序和运行结果再感受一下。

开始还是初始化，定位置，帮助指针找好位置，所有涉及指针的操作第一步反应都应该它指到哪里了。



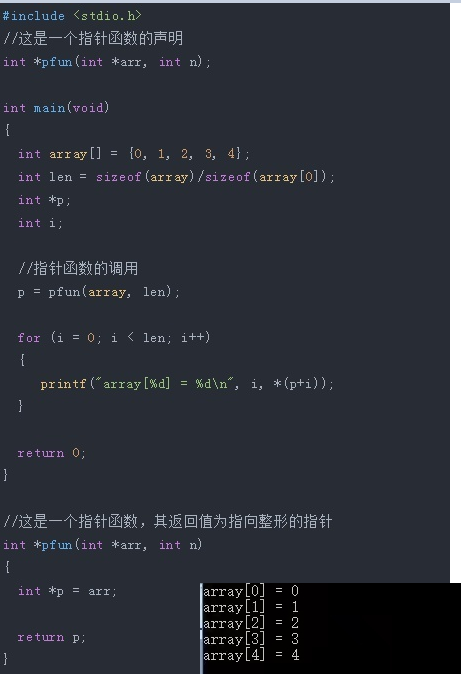
下面还是程序加执行结果来看一下。



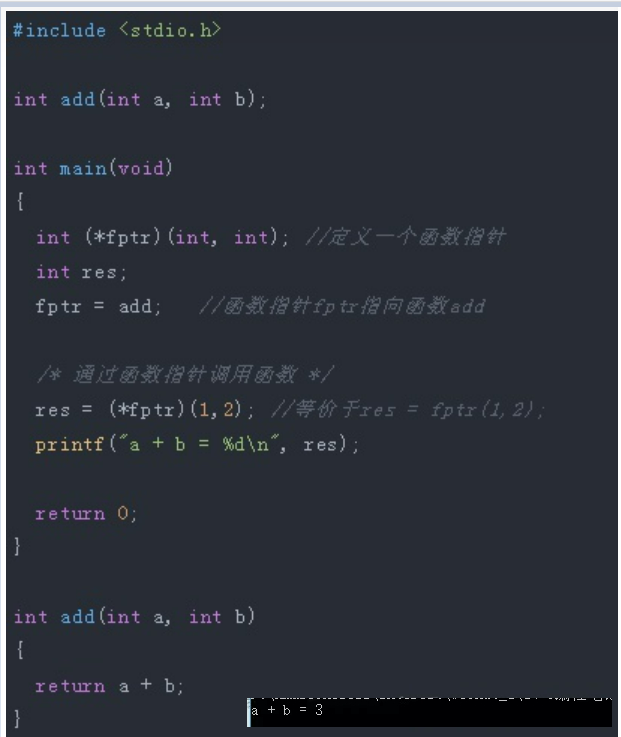
**4.函数指针和指针函数**

谈到这个问题，一样的肯定还是函数指针和指针函数比较应用一下。先来看指针函数。

指针函数指，针函数的本质是一个函数，其返回值是一个指针。示例如下：int \*pfun(int, int);由于“ \* ”的优先级低于“()”的优先级，因而pfun首先和后面的“()”结合，也就意味着，pfun是一个函数。即：int \*(pfun(int, int));接着再和前面的“\*”结合，说明这个函数的返回值是一个指针。由于前面还有一个int，也就是说，pfun是一个返回值为整型指针的函数。



函数指针其本质是一个指针变量，该指针变量指向一个函数。C程序在编译时，每一个函数都有一个入口地址，该入口地址就是函数指针所指向的地址。函数指针示例：func 是一个函数名，那么func与&func都表示的是函数的入口地址。同样的，在函数的调用中可以使用：方式一：func()，也可以使用方式二：(\*fun)()。这两种调用方式是等价的，只是我们平时大多都习惯用方式一的调用方法。至于为什么func与&func的含义相同：对于函数func来说，函数的名称就是函数代码区的常量，对它取地址(&func)可以得到函数代码区的地址，同时，fun本身也可以视为函数代码区的地址。因此，函数名称和对其取地址其含义是相同的。函数指针示例程序如下。



**四.文终**

本文尽于此，但指针的应用远不止这些，结构体的加入，链表的参与，熟悉只能从不断的编程，学习，项目，学习，编程反反复复。指针可以说是c的灵魂，愿每一个在面对它的你的灵魂可以与其早日会晤。Be be be patient。

**五.本文参考**

少部分例程有重复，感谢各位大佬的文章与分享，侵权删。

[c语言指针用法及实际应用详解，通俗易懂超详细！ - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/388456835)

[数组指针和指针数组的区别 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/55724052)

[指针函数，函数指针，有点懵圈了？你能分清楚吗 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/122355481)