**指针的深入理解，深刻掌握指针**

转载 2015年01月09日 16:39:59

* 标签：
* [指针](http://so.csdn.net/so/search/s.do?q=%E6%8C%87%E9%92%88&t=blog) /
* [c++](http://so.csdn.net/so/search/s.do?q=c++&t=blog) /
* [内存管理](http://so.csdn.net/so/search/s.do?q=%E5%86%85%E5%AD%98%E7%AE%A1%E7%90%86&t=blog)
* 974

﻿﻿

**指针的深入理解**

  “指针是什么？”“指针就是一种数据类型。”“你确定？”“那数据类型是什么？额，这个？？？类型就是类型，还能怎么解释嘛。”“指针有多少种？”“指针有好多种，比如指针，字符指针等等。”“指针是怎么确定分类的？”“就是声明时在变量类型后加个\*声明指针呀。”等等。这是我们对指针对基本的认识，可是指针到时是什么，什么是指针类型，指针的值和指针本身怎么区别，几乎初学者都凌乱了。给个简单的指针使用，倒还可以看明白，问点复杂的就懵了，到头来很委屈的回一句，“老师没教呀！”。  
     似乎初学者都有这样那样的无助，都寄希望与老师，可是自己没有基础，老师说了自己也听不懂，问了也白问，心里瘆的慌。问同学吧，也都和自己差不多，就算懂的也说不出个所以然，真是“茶胡子煮饺子”。  
    作为初学者到深入的我来说，没有老师的直接灌输，完全自己摸索成长，对于初学者的心理是完全明白，也知道为什么会有这么多的困惑。应试教育基本都是灌输，学生没有自主思考的机会，导致只能被动的接受，从而不能够深入。那么我将以自学者的角度深入分析指针，抛开书本中的条条框框，让你真正自由的学习指针的精髓。如果没有一个自由的心，将无法捕捉自由的指针的灵魂。  
    扯多了，进入正题。看完本文，相信第一次可能还不完全明白，但是一定要形成一种印象，然后多看几遍，在敲代码的过程中彻底理解，光靠看书是不可能完全理解的，一定要敲代码中思考才能掌握，但是文章会给你一个正确的方向，助你快速理解指针。  
    指针是一种数据类型，占4个字节。为什么指针的类型需要和具体的类型相关呢？比如，你不能说指针就是一种类型，而只能附带一种指向的类型来确定指针的类型。初学者很不理解这个意思，为什么指针还要依赖其他类型来确定类型呢？那么，如果你也这样问了，恭喜你，你慢慢上道了，不过，更可喜的是，本文可以加速你的思考，助你成功。现在，我们展开深入的分析了。下面请认真的阅读哦。  
    在计算机中，变量，指针，数据类型等等，都是反映在内存中的，当然这个内存指的是虚拟内存。虚拟内存，简单来说，就是我们程序使用的内存，不是真正的内存条内存，不过程序运行后，虚拟内存会映射到实际的内存即物理内存。了解虚拟内存，请阅读计算机操作内存章节。我们理解的程序中的内存都是虚拟内存，你当做是内存就是了。这个不清楚对我们程序理解没多少影响，有个概念就行了，继续往下看。  
    32位计算机只支持4GB内存大小，多出的是无效的，即使你装了8GB，也只能使用4GB，为什么最多只支持4GB呢，这就跟指针有关系了。计算机使用一个计算机字，32位的计算机就是32位，也就是4字节，因此这也就是为什么我们所知道的指针所占的内存是4字节的原因。因为我们的计算机普遍是32位的，也就是支持的最大内存是4GB，使用32位即4字节的指针就能够找到4GB的所有内存位置。如果将来4GB内存不够用，普及了64位的计算机，支持的内存大小就是2的64次方byte，远远超过了4GB，那么32位指针就找不到超出4GB以外的内存了，所以就不够了。因为32位指针所能表示的最大内存地址为2的32次方减一byte。所谓的不支持就是因为指针所能表示的范围没有内存的地址范围大，找不到全部的地址。  
    在目前的内存体系中，内存地址排序是线性的，也就是，32位地址（4GB内存）从地址0x00000000到0xFFFFFFFF 。因为是线性的体系结构，因此，内存地址都是按照序号递增的，如0,1,2,3,...，这样就表示了内存地址，当然，因为是用十六进制带前导0表示的。因此地址就是我们常见的0x00000000，0x00000001，0x00000002，0x00000003,...。正是因为是线性的，因此，根据这个序号就能够轻松的找到相应的内存地址，对内存进行读写。那么内存地址其实就是一个编号，从0开始的，指针所占内存的位数就决定了能表示的数字的大小，这个表示的数字就是内存地址。32位即4字节指针只能表示4GB内存，这也是我们现在知道的。所以，对于指针的认识，指针所占的大小就是这个意思，请反复揣摩。  
    那么我们在来看看数据类型。所谓的数据类型，在上我们理解成一种类型，但是本质的就是一个数据所占的的大小。比如占4字节，编译器把整型变量存放到内存中，就给一个4字节的内存来一个整数，这就是整型的意义。其他类型意义和这类似，当然除了指针。因为指针很特殊。指针的类型要依托其他的类型来定义，因为指针是找内存地址而存在的，因此，它也就不一般了。指针为4字节，也就是说，指针有能力走遍4GB空间，表明它有这个潜力，但是每次走多远确实根据指针的类型来定的，如果指针指向整型，那么指针的值每次递增，也就是指针表示的内存地址在递增，这个递增的跨度就是指针指向变量类型所占字节的大小。比如，指向整型的指针int \*p; p所指向的就是整型变量，所以p++;就是递增指针的值（内存地址）就是按照int型大小(4字节)每一次而增加的，如果开始指向的0字节，增加一次p的值就变成4字节了，再递增一次就是8字节了。而如果指向的是1字节的char \* pc，那么pc每次递增就只增加1字节，如果第一次是0字节，递增一次，就是1字节了。其他类型也是如此类似的。如果指针不根据数据类型来确定指针类型，那么指针就不知道如何递增，无法确定指针每次增加的跨度。如果使用4字节的指针去指向8字节的double类型变量，那么每次递增只能递增到double类型的一半，从而double类型的变量就被截断，从而读取的数据就错误了。所以这就是指针类型的意义。同样，这一段也要反复揣摩，虽然讲的很通俗，但是要把自己一直以来形成的思维冲破，需要努力几次，才能完全接受这种本质的理解。如果之前这样思考过，那么你就轻而易举的理解了。  
    把指针这样理解后，以后再来理解其他的，就能够游刃有余了。后续还会写指针与数组，指针常量和常量指针，内存读写保护和指针的操作关系，等等。但是，请先把这篇文章理解了，方可继续，否则还是会因为思维没扭转过来而疑惑重重。  
    如果有疑问，请在文章后提出，不然，自己不主动，永远都别想学好。我给大家提供一个机会，请好好把握。现在不是没人帮你学了，关键就看看你想不想沉下心学下去。如果能够做到，你将会很快成为高手。  
    提醒一下，上面所讲述的内容，我是结合了，管理，计算机组成原理，C/C++程序设计等知识的综合，所以讲述的深入浅出，如果自己一下子理解不透彻，请勿沮丧，那是要有一定的基础才能够理解透彻的。只要能够从本文中扭转自己的思维，你就成功了。思维才是最重要的，当然，如果你能够全部吸收了，那就更要恭喜你了。助你成功，好好加油！！