

请问指针和地址区别和联系是什么，内存中的数据具体是怎么参与运算？

<https://www.zhihu.com/question/357298029/answer/908027208>

我觉得你这个问题，根本就不是一个问题。

“指针和地址的区别和联系”比较好回答，这个我下面说。“内存如何参与运算”至少要谈到计算机组成与结构、操作系统相关课程，如果要深入到“物理上的过程”又必须涉及数电，这个问题差不多已经把计算机专业本科主要课程都涵盖了，从不同层次讲几天几夜也讲不完。那题主问这个问题应该是还刚刚接触这个专业，可能也没有计组和操作系统的的基础知识，后一个问题理解起来也十分有难度，我就不说了（实际上我也是懒，内容太多，梳理成通俗易懂实在太困难）。

我就简单说一下内存地址和指针的关系吧。

很多C或者C++的入门教材会说指针和地址是同一个东西，这样说有一定的道理，主要是避免了给初学者带来过于冗长且难懂的讨论，但是解释的非常局限，解释力也很差。当涉及到多种语言，以及当“指针”和“引用”放在一起讨论时，这种解释就会引起很多混乱。

要说指针和地址的区别和联系，首先要理解“变量”的概念。变量是一个综合的结构，在最简单的理解中它至少包含两部分，“变量名”和“值”。比如在C语言中写下这行代码

```
int a = 1;
```

你就声明并且定义了一个变量，它的变量名是 a 值是整数 1。这个值需要存储在内存的某个地方，需要占一定的空间，有人会说“这个内存地址就是指向变量a的指针”，这也是很多教材的解释，但其实不是这样简单。在继续之前我们继续来看“变量”这个概念。上面我们提到了一个变量至少有“变量名”和“值”两个部分，但显然不止这些，如果我们看下面这两行代码

```
int a = 1;
int b = 1;
```

我们这里有两个变量，a和b，他们名字不同，但是值是相同的，都是整数 1。我们能不能说他们是同一个变量呢？显然不能。但是他们好像除了名字以外都完全相同。如果你仔细想，你会发现这种矛盾的根本是在于“一样”和“同一”是两个概念，这两个变量存储的内容是“一样的”，但不是“同一的”。所以当我们说一个变量存储了什么，并不能简单地用“值”来表示，同一个“值”可以被存储在不同的地方，他们一样但不同一，你可以说把“地址”引入作为变量的一部分，但是我喜欢用更通用的说法“对象”。所以一个变量实际上应该由“变量名”和“对象”构成，“值”是对象的属性，不是变量的。

可以通过这样一个例子来理解这三个概念。我的数学成绩和你的数学成绩都是90分，那么这两个成绩“本身”，就是两个不同的成绩“对象”。为了方便指代，我们可以用“A的成绩”和“B的成绩”这两个名字来代指两个成绩，这就是变量名。而成绩这个对象里面有一些属性，比如90这个数值是用分数量化的成绩，这些属性和值的集合就是这个对象的“值”，在成绩这个例子里就是整数90。

虽然教材一般都会先说C或者C++都有的“指针”，但是我想先说“引用”，因为引用虽然C没有，但是C++有、java有、python有、js有，几乎所有现代程序设计语言都有。我这里说的“引用”都是指“左值引用”，也就是“别名”。比如下面两行C++代码

```
int a = 1;
int &b = a;
```

这得到了什么？一个变量a和变量b，变量a的变量名是“a”，对象是一个值为1的整数。b呢？b的变量名是“b”，b表示的对象与变量a表示的对象是同一个对象。所以两个变量，只是变量名不同，但是指代的对象是同一个。就好像A说“我的成绩”和“A的成绩”，其实是同一个东西，只是叫法不同。所以上面两行定义之后，对b做的所有操作，实际上都是对a做的操作。这个解释对java和python都完全适用。

说完了引用可以说指针了。我们先看下面两行 C的代码

```
int a = 1;
int *b = &a;
```

在这里，变量a和变量b是完全不同的两个变量。他们的变量名和对象都完全不同。变量a就不解释了。变量b的变量名是“b”，对象是一个“指向变量a所指代对象的指针”。我没有用“变量b存储的就是a的地址”这种浅显的说法，而是用了“指向变量a所指代对象的指针”这句有点拗口的说法，这是有好处的，这样解释力更强，更统一。我们把这句“指向变量a所指代对象的指针”拆开看看，先来看看“变量a所指代对象”，变量a所指代对象就是一个值为1的整数。然后再看“指向xxx的指针”，这是一种有趣的对象，它可以通过某种方法“获取”所指向对象。就好像“成绩登记表”和“成绩”，显然不是同一种东西，但是成绩登记表“指向”了某些成绩，通过这种“指向”的关系，成绩登记表可以读取某个成绩，可以修改某个成绩。所以“指向某对象”在这里是表示“具有获取某对象的能力”。所以b是一个“指向变量a所指代对象的指针”，表明b所指代对象具有获取变量a所指代对象的能力。这是指针变量和普通变量的关系。

指针和引用十分不一样，指针变量所指代对象具有获取所指向对象的能力，引用变量所指代对象与被引用变量所指代对象是同一对象。所以一个是具有指向能力的对象，一个是“别名”，两个完全不同。

那么再来看指针和地址的区别和联系。我们说一个指针对象“具有获取被指向对象的能力”，但是没有说如何获取。一种简单的方法就是通过内存“地址”，任何对象都是数据实体，需要存储在内存某个地方，这个地方有地址。所以根据一个“对象所在地址”就可以找到这个对象，进而获取这个对象本身。所以指针最简单的一种实现方法就是，“指针对象记录被指向对象的地址”，实际上目前C语言就是这么实现的。

但是我们不能说“指针就是地址”，“指针对象记录被指向对象的地址”只是一种具体的实现方法。而“指针”是程序设计语言的概念，语法概念不应该被具体实现方法所定义。“指针”的个语法概念的定义就应该是类似于“指针具有获取所指向对象的能力”这样，而“指针对象记录被指向对象的地址”是一种可行的实现，也是目前广泛使用的实现。但实际上我们可以有不同实现，比如“指针对象记录一个指代被指向对象的变量名”，这种实现在静态语言中也是可行的。而且“指针对象记录被指向对象的地址”也不是一定可行的实现，比如对于某些有更灵活的内存管理的语言来说，如果来实现“指针”这个概念，就不能使用记录地址这种方法了，因为一个对象在生命周期中其地址可能会改变。

所以总结来说，指针和地址没有关系，指针就是指针，地址就是地址，硬要说有关系就是，地址可以被用来实现指针。

说点题外话，其实很多C、C++语言的教材，为了方便，都没有很详细地说明白“变量”是什么。就说变量是一个空间，里面可以存储东西。然后提到指针就说指针存的是变量的地址。这如果是对于已经理解的人来说，的确也没有特别大的错误，因为目前C的实现就是这样的，把变量名、对象、和存储的位置绑定了，所以就没必要区分这三个东西了，然后指针也的确就是存一个地址，通过地址获取所指向对象。但是这种解释看起来很混乱、不统一、而且把语言的设计和语言的具体实现混为一谈。当要介绍更高级的功能，比如“引用”，时又搬出不一样的解释，没法跟前面对指针的说明统一起来。给初学者带来了极大的混乱。

- 评论

- C语言中，“指针”这个概念说的就是地址，指针就是地址，地址是一个数值，用16进制来表示比如就是这个样子：0X111FFFF 那么谁来保存这个0X111FFFF？那就是指针变量，比如int *pInt，指针变量pInt的作用就是用来保存这个地址的。答主应该是说指针变量（简称指针）不是地址吧。
- 变量名和地址其实都是对于某一个内存单元的唯一标识符，前者适合于程序员理解，后者适用于机器理解，但是变量名最终还是通过地址来实现对于一个内存单元的唯一标识的，这样理解 是合理的么？
- 请问一下，对于二维数组a[2][2]，其中a[0]和a[0][0]的地址是一致的，然后对于数组a[0]和最小元素a[0][0]进行取值得到的结果是不一致的，前者是一个地址，后者是a[0][0]的值。*(&a[0]), *(&a[0][0])的值不是一样的。当然可以理解成a[0]存储的是一个一维数组的地址。但是二者地址相同，存储的值确不是一样的，这里应该怎么理解呢？或者&a[0]和&a[0][0] 不是常量么？