

C语言笔记第一章指针的概念 PDF转换可能丢失图片或格式  
，建议阅读原文

[https://www.100test.com/kao\\_ti2020/135/2021\\_2022\\_C\\_E8\\_AF\\_AD\\_E8\\_A8\\_80\\_E7\\_AC\\_94\\_c97\\_135138.htm](https://www.100test.com/kao_ti2020/135/2021_2022_C_E8_AF_AD_E8_A8_80_E7_AC_94_c97_135138.htm) 第一章 指针的概念

指针是一个特殊的变量，它里面存储的数值被解释成为内存里的一个地址。要搞清一个指针需要搞清指针的四方面的内容：指针的类型，指针所指向的类型，指针的值或者叫指针所指向的内存区，还有指针本身所占据的内存区。让我们分别说明。先声明几个指针放着做例子：例一：(1)int \*ptr.

(2)char \*ptr. (3)int \*\*ptr. (4)int (\*ptr)[3]. (5)int \*(\*ptr)[4]. 如果看不懂后几个例子的话，请参阅我前段时间贴出的文章 的复杂类型声明>>。

1. 指针的类型。从语法的角度看，你只要把指针声明语句里的指针名字去掉，剩下的部分就是这个指针的类型。这是指针本身所具有的类型。让我们看看例一中各个指针的类型：(1)int \*ptr. //指针的类型是int \* (2)char \*ptr. //指针的类型是char \* (3)int \*\*ptr. //指针的类型是 int \*\* (4)int (\*ptr)[3]. //指针的类型是 int(\*)[3] (5)int \*(\*ptr)[4]. //指针的类型是 int \*(\*)[4] 怎么样？找出指针的类型的方法是不是很简单？

2. 指针所指向的类型。当你通过指针来访问指针所指向的内存区时，指针所指向的类型决定了编译器将把那片内存区里的内容当做什么来看待。从语法上看，你只须把指针声明语句中的指针名字和名字左边的指针声明符 \*去掉，剩下的就是指针所指向的类型。例如：(1)int \*ptr. //指针所指向的类型是int (2)char \*ptr. //指针所指向的的类型是char (3)int \*\*ptr. //指针所指向的的类型是 int \* (4)int (\*ptr)[3]. //指针所指向的的类型是 int()[3] (5)int \*(\*ptr)[4]. //指针所指向的的类型是 int

\*()[4] 在指针的算术运算中，指针所指向的类型有很大的作用。指针的类型(即指针本身的类型)和指针所指向的类型是两个概念。当你对C越来越熟悉时，你会发现，把与指针搅和在一起的“类型”这个概念分成“指针的类型”和“指针所指向的类型”两个概念，是精通指针的关键点之一。我看了不少书，发现有些写得差的书中，就把指针的这两个概念搅在一起了，所以看起书来前后矛盾，越看越糊涂。

3. 指针的值，或者叫指针所指向的内存区或地址。指针的值是指针本身存储的数值，这个值将被编译器当作一个地址，而不是一个一般的数值。在32位程序里，所有类型的指针的值都是一个32位整数，因为32位程序里内存地址全都是32位长。指针所指向的内存区就是从指针的值所代表的那个内存地址开始，长度为sizeof(指针所指向的类型)的一片内存区。以后，我们说一个指针的值是XX，就相当于说该指针指向了以XX为首地址的一片内存区域；我们说一个指针指向了某块内存区域，就相当于说该指针的值是这块内存区域的首地址。指针所指向的内存区和指针所指向的类型是两个完全不同的概念。在例一中，指针所指向的类型已经有了，但由于指针还未初始化，所以它所指向的内存区是不存在的，或者说是无意义的。以后，每遇到一个指针，都应该问问：这个指针的类型是什么？指针指向的类型是什么？该指针指向了哪里？

4. 指针本身所占据的内存区。指针本身占了多大的内存？你只要用函数sizeof(指针的类型)测一下就知道了。在32位平台里，指针本身占据了4个字节的长度。指针本身占据的内存这个概念在判断一个指针表达式是否是左值时很有用。

100Test 下载频道开通，各类考试题目直接下载。详细请访问

