

C语言笔记第八章指针类型转换 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_135100.htm 第八章 指针类型转换

当我们初始化一个指针或给一个指针赋值时，赋值号的左边是一个指针，赋值号的右边是一个指针表达式。在我们前面所举的例子中，绝大多数情况下，指针的类型和指针表达式的类型是一样的，指针所指向的类型和指针表达式所指向的类型是一样的。例十四：1。float f=12.3。2。float *fptr=&f。不对。因为指针p的类型是int*，它指向的类型是int。表达式&f。如果有一个指针p，我们需要把它的类型和所指向的类型改为TYPE*和TYPE，那么语法格式是：(TYPE*)p；这样强制类型转换的结果是一个新指针，该新指针的类型是TYPE*，它指向的类型是TYPE，它指向的地址就是原指针指向的地址。而原来的指针p的一切属性都没有被修改。一个函数如果使用了指针作为形参，那么在函数调用语句的实参和形参的结合过程中，也会发生指针类型的转换。例十五：void fun(char*). int a=125,b。fun((char*)&a)的结果是一个指针，它的类型是int*，它指向的类型是int。形参这个指针的类型是char*，它指向的类型是char。这样，在实参和形参的结合过程中，我们必须进行一次从int*类型到char*类型的转换。结合这个例子，我们可以这样来想象编译器进行转换的过程：编译器先构造一个临时指针char*temp，然后执行temp=(char*)&a，最后再把temp的值传递给s。所以最后的结果是：s的类型是char*，它指向的类型是char，它指向的地址就是a的首地址。 100Test 下载频道开通，各类考试题目

直接下载。详细请访问 www.100test.com