

C语言笔记第二章指针的算术运算 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\\_135145.htm](https://www.100test.com/kao_ti2020/135/2021_2022_C_E8_AF_AD_E8_A8_80_E7_AC_94_c97_135145.htm) 第二章 指针的算术运算 指针可以加上或减去一个整数。指针的这种运算的意义和通常的数值的加减 运算的意义是不一样的。例如：例二：1。char a[20]. 2。int \*ptr=a. .... 3。ptr. 在上例中，指针ptr的类型是int\*,它指向的类型是int，它被初始化为指向整形变量a。接下来的第3句中，指针ptr被加了1，编译器是这样处理的：它把指针 ptr的值加上了sizeof(int)，在32位程序中，是被加上了4。由于地址是用字节做单位的，故ptr所指向的地址由原来的变量a的地址向高地址方向增加了4个字节。由于char类型的长度是一个字节，所以，原来ptr是指向数组a的第0号单元开始的四个字节，此时指向了数组a中从第4号单元开始的四个字节。我们可以用一个指针和一个循环来遍历一个数组，看例子：例三：例三：int array[20]. int \*ptr=array. ... //此处略去为整型数组赋值的代码。 ... for(i=0;i{ (\*ptr) . ptr ; } 这个例子将整型数组中各个单元的值加1。由于每次循环都将指针ptr加1，所以每次循环都能访问数组的下一个单元。再看例子：例四：1。char a[20]. 2。int \*ptr=a. .... 3。ptr =5. 在这个例子中，ptr被加上了5，编译器是这样处理的：将指针ptr的值加上5乘sizeof(int)，在32位程序中就是加上了5乘4=20。由于地址的单位是字节，故现在的ptr所指向的地址比起加5后的ptr所指向的地址来说，向高地址方向移动了20个字节。在这个例子中，没加5前的ptr指向数组a的第0号单元开始的四个字节，加5后，ptr已经指向了数组a的合法范围之

外了。虽然这种情况在应用上会出问题，但在语法上却是可以的。这也体现出了指针的灵活性。如果上例中，ptr是被减去5，那么处理过程大同小异，只不过ptr的值是被减去5乘sizeof(int)，新的ptr指向的地址将比原来的ptr所指向的地址向低地址方向移动了20个字节。总结一下，一个指针ptrold加上一个整数n后，结果是一个新的指针ptrnew，ptrnew的类型和ptrold的类型相同，ptrnew所指向的类型和ptrold所指向的类型也相同。ptrnew的值将比ptrold的值增加了n乘sizeof(ptrold所指向的类型)个字节。就是说，ptrnew所指向的内存区将比ptrold所指向的内存区向高地址方向移动了n乘sizeof(ptrold所指向的类型)个字节。一个指针ptrold减去一个整数n后，结果是一个新的指针ptrnew，ptrnew的类型和ptrold的类型相同，ptrnew所指向的类型和ptrold所指向的类型也相同。ptrnew的值将比ptrold的值减少了n乘sizeof(ptrold所指向的类型)个字节，就是说，ptrnew所指向的内存区将比ptrold所指向的内存区向低地址方向移动了n乘sizeof(ptrold所指向的类型)个字节。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](http://www.100test.com)