

C语言程序设计(第6章指针)3 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_A8_8B_c97_135156.htm 6.3 指针运算符与指针表达式

6.3.1 指针运算符与指针表达式

在C中有两个关于指针的运算符：
&运算符：取地址运算符，#8226.
运算符：指针运算符， ptr表示其所指向的变量。

[例6-2] 从键盘输入两个整数，按由大到小的顺序输出。

```
main( ){ int *p1,*p2,a,b,t; /* 定义指针变量与整型变量 */ scanf("%d,%d",&a,&b); p1 = &a; p2 = &b; if (*p1 > *p2) { /* 交换指针变量指向的整型变量 */ t = *p1; *p1 = *p2; *p2 = t; } printf("%d,%d\n",a,b); }
```

在程序中，当执行赋值操作 `p1 = &a` 后，指针实实在在地指向了变量 `a` 与 `b`，这时引用指针 `*p1` 与 `*p2`，就代表了变量 `a` 与 `b`。运行程序：3, 4 在程序运行过程中，指针与所指的变量之间的关系如图6-4所示：当指针被赋值后，其在内存的安放如a)，当数据比较后进行交换，这时，指针变量与所指向的变量的关系如b)所示，在程序的运行过程中，指针变量与所指向的变量其指向始终没变。下面对程序做修改。

[例6-3]

```
main( ){ int *p1,*p2,a,b,*t; scanf("%d,%d",&a,&b); p1 = &a; p2 = &b; if (*p1 < *p2) { /* 指针交换指向 */ t = p1; p1 = p2; p2 = t; } printf("%d,%d\n",*p1,*p2); }
```

程序的运行结果完全相同，但程序在运行过程中，实际存放在内存中的数据没有移动，而是将指向该变量的指针交换了指向。其示意如图6-5：当指针交换指向后，`p1`和`p2`由原来指向的变量`a`和`b`改变为指向变量`b`和`a`，这样一来，`*p1`就表示变量`b`，而`*p2`就表示变量`a`。在上述程序中，无论在何时，只要指针与所指向的变量满足 `p = &a`，`*p`等效于变量`a`。

6.3.2 指针

变量作函数的参数 函数的参数可以是我们在前面学过的简单数据类型，也可以是指针类型。使用指针类型做函数的参数，实际向函数传递的是变量的地址。由于子程序中获得了所传递变量的地址，在该地址空间的数据当子程序调用结束后被物理地保留下来。[例6-4] 利用指针变量作为函数的参数，用子程序的方法再次实现上述功能。

```
main( ){ void chang(). /*  
函数声明*/ int *p1,*p2,a,b,*t. scanf("%d, %d", amp.b). p1 =  
amp.b. chang(p1 , p2). /*子程序调用*/ printf("%d, %d\n" , *p1,  
*p2). return 0.}void chang(int *pt1,int *pt2){ /*子程序实现将两数值  
调整为由大到小*/ int t. if (*pt1 { t=*pt1. *pt1=*pt2. * pt2 = t. }  
return.}
```

由于在调用子程序时，实际参数是指针变量，形式参数也是指针变量，实参与形参相结合，传值调用将指针变量传递给形式参数pt1和pt2。但此时传值传递的是变量地址，使得在子程序中pt1和pt2具有了p1和p2的值，指向了与调用程序相同的内存变量，并对其在内存存放的数据进行了交换，其效果与[例6 - 2]相同。思考下面的程序，是否也能达到相同的效果呢？

```
main( ){ void chang(). int *p1,*p2,a,b,*t.  
scanf("%d,%d", amp.b). p1 = amp.b. chang(p1, p2). printf("%d,  
%d\n" , *p1, *p2).}void chang(int *pt1,int *pt2){ int *t. if (*pt1 {  
t=pt1. pt1=pt2. p t 2 = t. } return.}
```

程序运行结束，并未达到预期的结果，输出与输入完全相同。其原因是对子程序来说，函数内部进行指针相互交换指向，而在内存存放的数据并未移动，子程序调用结束后，main()函数中p 1和p 2保持原指向，结果与输入相同。

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