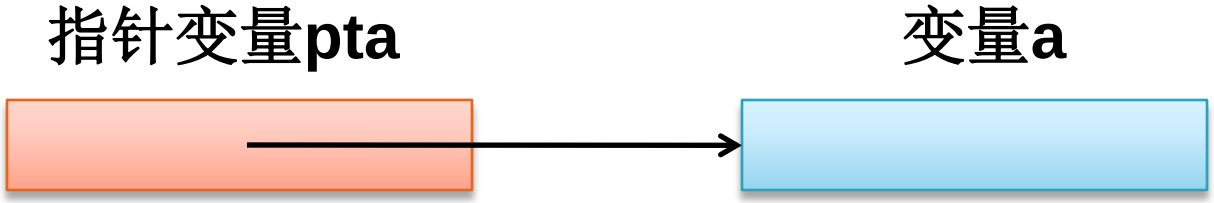
2.变量与指针

杨振平

变量与指针变量

- 变量有地址，指针变量可以存放变量的地址。
- 当指针变量中存放某个变量的地址后，我们就说该指针变量指向这个变量。
- 如：指针变量pta指向变量a



如何通过pta访问它所指向的变量a?

使用指针变量的三个基本步骤

（1）定义指针变量

即给指针变量分配内存空间。

（2）对指针变量赋值

即使指针变量指向某对象，该对象可以是变量、数组、函数或动态分配的一块内存空间等。

（3）通过指针变量间接访问所指向的对象。

定义指针变量

指针变量的定义格式：

数据类型　*变量名；

其中：

　*－是指针类型变量的标志符号。

　变量名－为指针变量名（构成同标识符）。

　数据类型－为指针变量所指向变量的数据类型。

数据类型　*　；表示指针类型

说明：定义一个指针变量，系统将为该指针变量分配一定大小的内存（在Dev-C++中，每个指针变量占有8个字节的长度）。

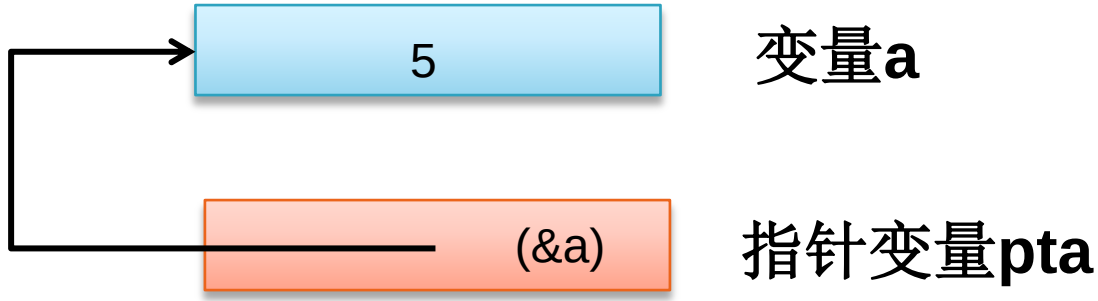
指针变量的初始化

在定义指针变量的同时为指针变量提供初值。

如： int a=5,*pta=&a;

其中a的初值为5，pta的初值为整型变量a的地址。

这时，pta与a的关联如下：



即，指针变量**pta**指向变量**a**。 pta ⟶ a

使用赋值语句为变量提供初值

上述定义语句： int a=5,*pta=&a;

与下面语句组的功能是等效的。

```
int a,*pta; //先定义变量
a=5;        //使用赋值语句提供初值
pta=&a; //使用赋值语句提供初值
```

提醒： pta=&a; 不可写成：*pta=&a;

因为， *pta并不表示指针变量pta，而表示pta所指向的变量a。

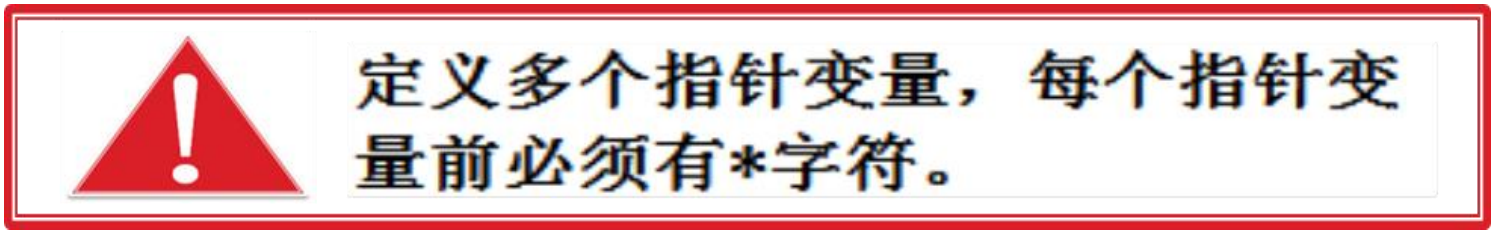
指针变量与指针变量所指向的变量是两个完全不同的概念。

定义多个指针变量

例如： double *p1,*p2;

定义2个双精度型的指针变量p1和p2，它们只能指向double型变量。

变量p1和p2的类型为：double *。



通过指针变量间接访问所指向的变量

- 指针类型中有两个特殊的单目运算符：&和*。

（1）&–取地址运算符

&变量名　//获取变量的内存单元地址

（2）*–指针运算符（也称为间接访问运算符）

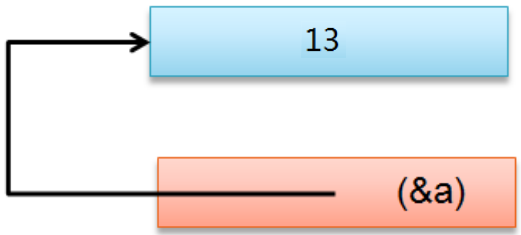
*指针变量名 或 *指针常量　//表示该指针所指向的变量

如果指针变量pta中存放着变量a的指针，则*pta表示pta所指向的变量即变量a 。这是一种间接访问的表示。

通过指针变量间接访问所指向的变量(续)

- 如： int a=5,*pta=&a;
  *pta=a+8;
  cout<<a<<“,”<<*pta<<endl;
- 变量初始分配示意图如下：

在这里，*pta是表示pta所指的对象，即变量a。



变量a

指针变量pta

*pta等同于变量a

输出结果：13，13

分析下面程序中各输出项的意义

```
int a=5,*p=&a;
cout<<&a<<endl;     //a的地址
cout<<a<<endl;      //a的值
cout<<&p <<endl;    //p的地址
cout<<p<<endl;      //p的值
cout<<*p<<endl;     //p所指变量的值
```

```
0x23fe4c
5
0x23fe40
0x23fe4c
5
```

注意：指针变量的值一定是“地址”；指针变量所指对象的值不一定是“地址”。

使用指针变量时应注意的几点

1. 不要访问没有被初始化的指针变量。

   如：int *p;

   cin>>*p;　　这样使用指针p是危险的！

   由于p变量未初始化，p中可能存在一个不确定的单元地址，这时的输入将会改变原存储单元的值，造成结果混乱。

2. 指针变量可以有空值，即该指针变量不指向任何变量。

   常用符号常量NULL表示空指针值，其实NULL代表的值是整数0。编译系统约定0号单元不存放有效数据。