

4.数组与指针

杨振平

一维数组的指针

- ▶ 数组占据内存中一块连续的存储空间，**每个数组元素都有确定的内存地址**；可通过定义指向数组元素类型的指针变量，间接访问数组中的各个元素。
- ▶ C++语言规定，**数组名代表数组的首地址**（即数组中第一个元素的地址），它是一个常量指针。

如：int a[10], *p=a;

说明a是一个整型数组，p是一个整型的指针变量，且p指向a数组，其中a为数组名，代表数组的首地址，即&a[0]。

指针类型的算术和关系运算

► 指针的算术运算

(1) 指针 \pm 整数 \Rightarrow 指针

假设: `int a[10]={10, 20, 30}, *p=a, i;`

`p+i`: 表示`p`所指元素之后的第`i`个元素的指针

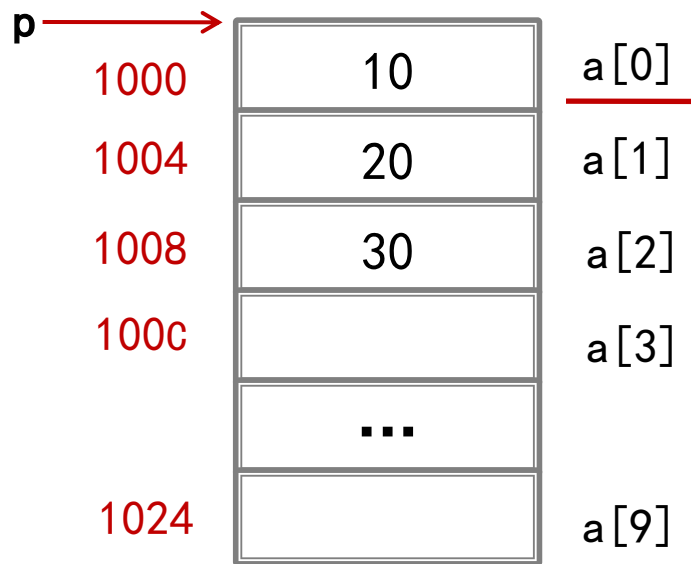
`p-i`: 表示`p`所指元素之前的第`i`个元素的指针

`cout<<*p<<endl;` 结果: 10

`p++;` // `p`指向`a[1]`

`cout<<*p<<endl;` 结果: 20

`p++;` // `p`指向`a[2]`



a数组内存分配示意图

指针类型的算术和关系运算

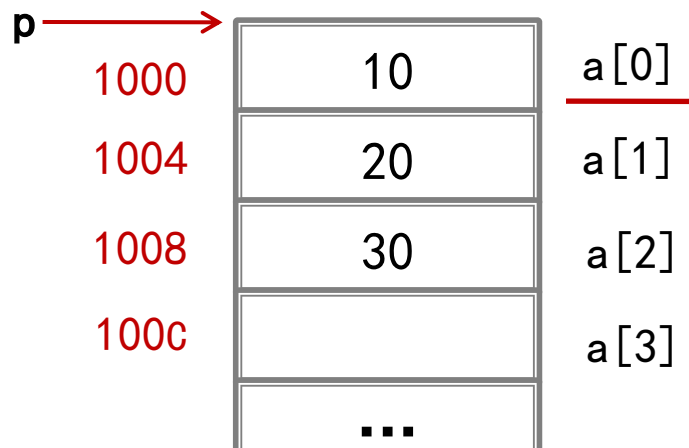
► 指针的算术运算

(1) 指针 ± 整数 == 》 指针

假设: `int a[10]={10, 20, 30}, *p=a, i;`

`p+i`: 表示`p`所指元素之后的第`i`个元素的指针

`p-i`: 表示`p`所指元素之前的第`i`个元素的指针



1001。

指针的算术运算与数学中运算不同。如 `p` 的初值为 `a[0]` 的地址，即 1000，执行 `p++` 后，`p` 将指向 `a[1]` 元素，这时 `p` 的值为 1004，而不是

实际上，在 C++ 中指针的算术运算与指针指向的变量类型有关，如 `p` 指向 `int` 型，因 `int` 型变量为 4 字节长度，所以，`p+1` 相当于 `p+4`，同样 `p+i` 相当于 `p+i*4`。

指针类型的算术和关系运算

(2) 两个同类型的指针做减法运算

指针2-指针1 ==> 整数

常用于计算两个指针之间包含元素的个数。计算方法:

$$\frac{\text{指针2} - \text{指针1}}{\text{元素的字节长度}}$$

如: p1,p2是整型指针, p1中的地址为1000, p2中的地址为1008

那么p2-p1, 相当于(1008-1000)/4, 结果为2, 说明p1到p2之间包含2个元素。