

---

# 嵌入式开发

## 指针（二）

---

创客学院 小美老师

▶ 1 / 掌握指针的运算

2 / 总结与思考

# 指针运算

- 指针运算是以指针变量所存放的地址量作为运算量而进行的运算
- 指针运算的实质就是地址的计算
- 指针运算的种类是有限的，它只能进行赋值运算、算术运算和关系运算

# 指针运算

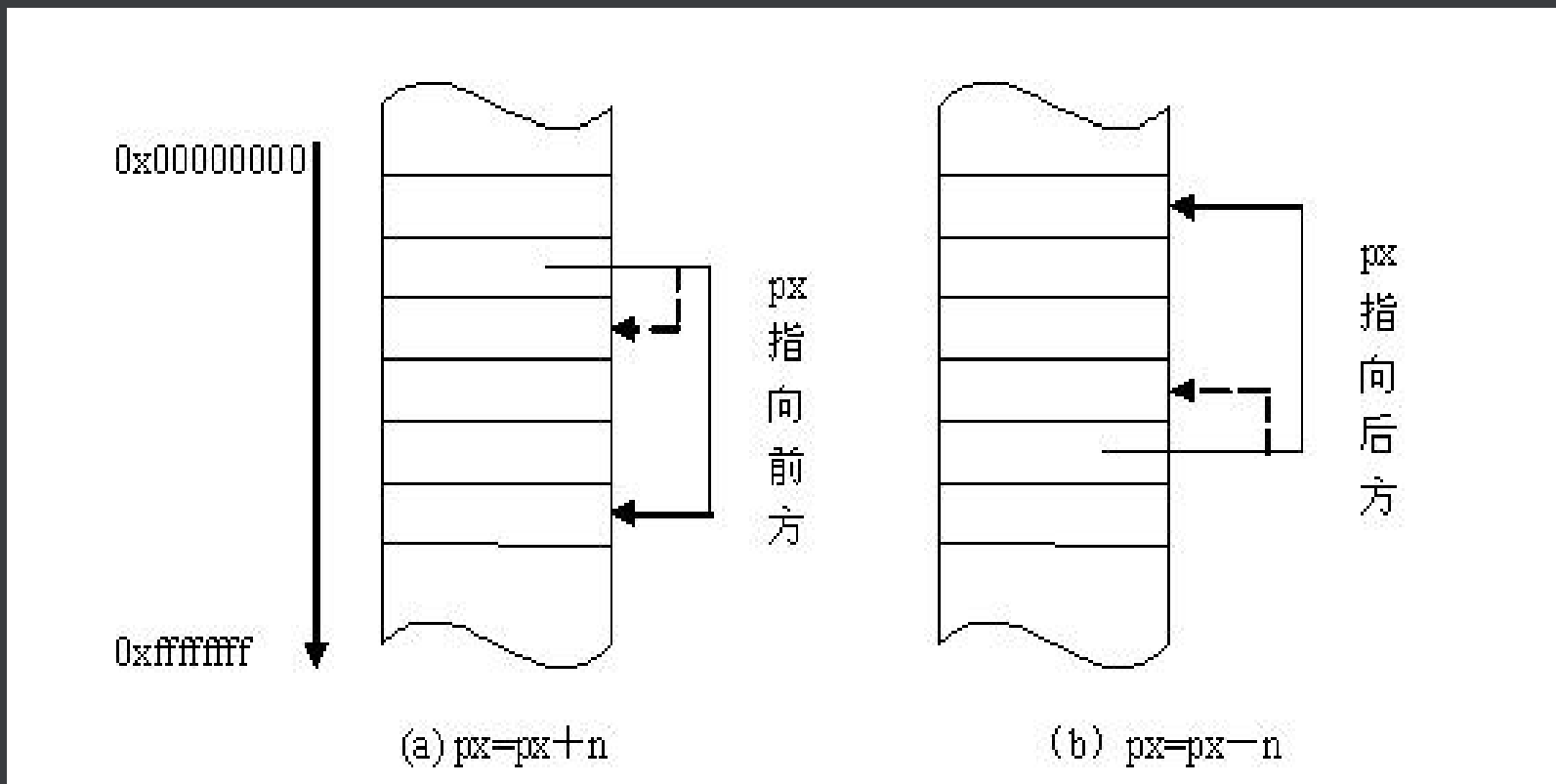
指针的算术运算见下表：

运算符	计算形式	意 义
+	$px+n$	指针向地址大的方向移动 $n$ 个数据
-	$px-n$	指针向地址小的方向移动 $n$ 个数据
++	$px++$ 或 $++px$	指针向地址大的方向移动 1 个数据
--	$px--$ 或 $--px$	指针向地址小的方向移动 1 个数据
-	$px-py$	两个指针之间相隔数据元数的个数

# 指针运算

指针加减一个n的运算:  $px + n$

$px - n$



# 指针运算

## ➤ 注意

- 不同数据类型的两个指针实行加减整数运算是无意义的
- $px+n$ 表示的实际位置的地址量是：
  - $(px) + \text{sizeof}(px\text{的类型}) * n$
- $px-n$ 表示的实际位置的地址量是：
  - $(px) - \text{sizeof}(px\text{的类型}) * n$

# 指针运算

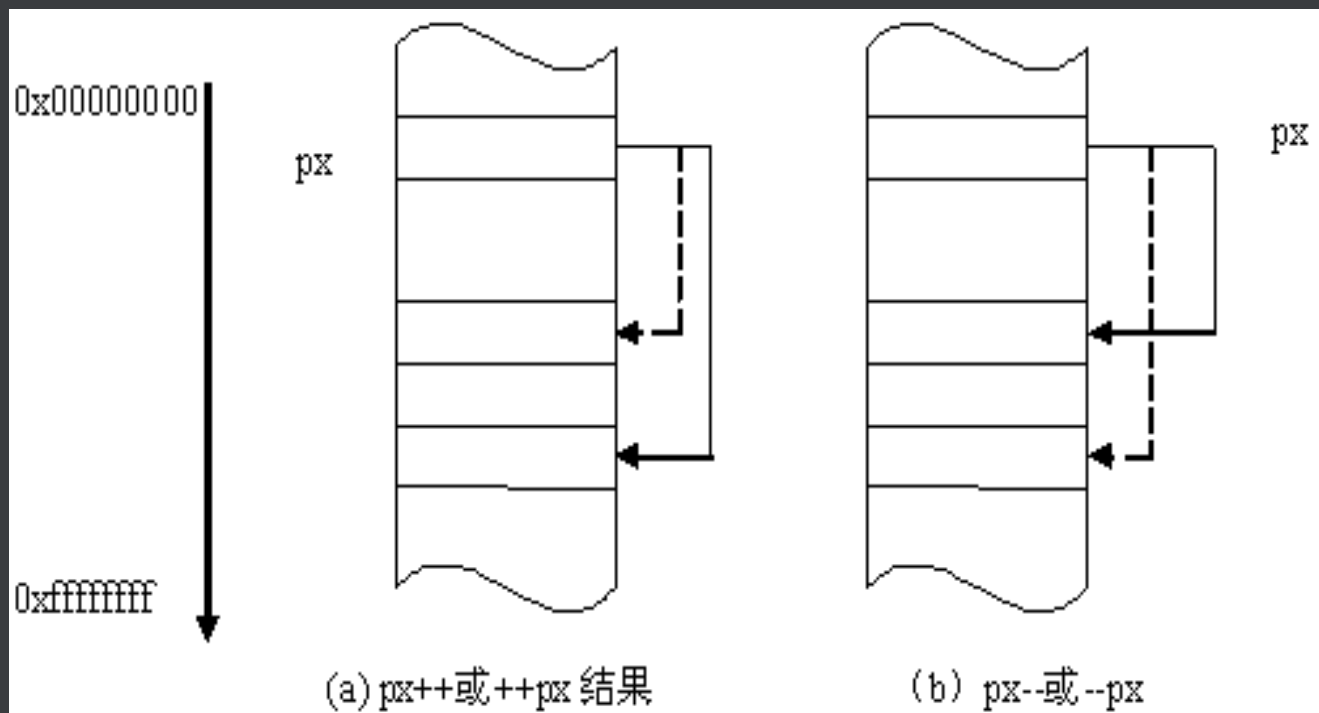
## ➤ 两指针相减运算

- $px - py$  运算的结果是两指针指向的地址位置之间相隔数据的个数。因此，两指针相减不是两指针持有的地址值相减的结果。
- 两指针相减的结果值不是地址量，而是一个整数值，表示两指针之间相隔数据的个数。

# 指针运算

## 指针加一、减一运算

$px++$ ,  $++px$ ,  $px--$ ,  $--px$





# 指针运算

## 指针的关系运算符

表 4.2 指针的关系运算符

运算符	说明	例子
>	大于	<code>px &gt; py</code>
<	小于	<code>px &lt; py</code>
>=	大于等于	<code>px &gt;= py</code>
<=	小于等于	<code>px &lt;= py</code>
!=	不等于	<code>px != py</code>
==	等于	<code>px == py</code>

# 指针运算

## 指针关系运算

- 两指针之间的关系运算表示它们指向的地址位置之间的关系。指向地址大的指针大于指向地址小的指针。
- 指针与一般整数变量之间的关系运算没有意义。但可以和零进行等于或不等于的关系运算，判断指针是否为空。

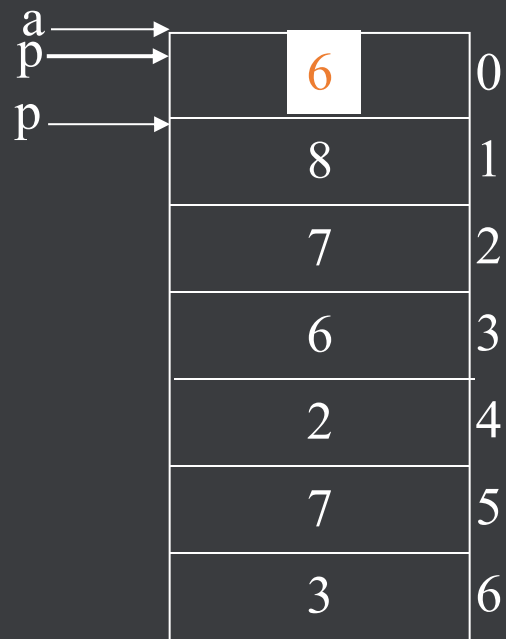
# 程序举例

```
int main()
{
    int a[]={5,8,7,6,2,7,3};
    int y,*p=&a[1];

    y=(*--p)++;

    printf("%d ",y);
    printf("%d",a[0]);
}
```

输出： 5 6

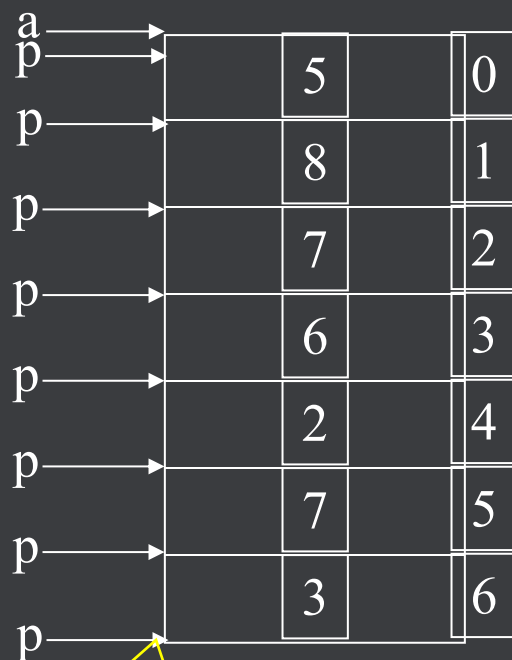


# 程序举例

## 注意指针的当前值

```
int main()
{
    int i, *p, a[7];
    p = a;
    for(i = 0; i < 7; i++)
        scanf("%d", p++);
    printf("\n");

    p = a;
    for(i=0; i<7; i++) {
        printf("%d", *p);
        p++;
    }
}
```



指针变量可以指到数组后的内存单元

# 总结与思考

主要介绍指针的各种运算，包括算术元素和关系运算

思考

- 指针运算的本质是什么？
- 指针加1，移动多少字节？

扫一扫，获取更多信息



THANK YOU