

# C语言

## 第9讲：指针的应用

赵岩

哈尔滨工业大学软件学院

August 20, 2011

# 目录

## ① 指针和函数

# 目录

- 1 指针和函数
- 2 指针和一维数组

# 目录

- 1 指针和函数
- 2 指针和一维数组
- 3 字符型指针

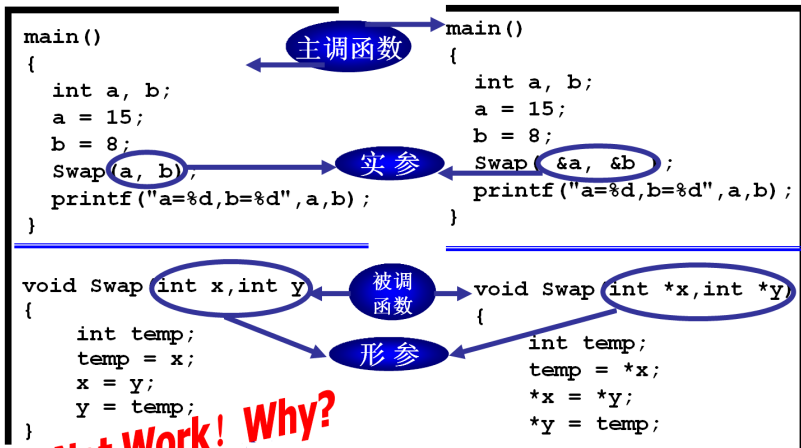
# 指针参数

- 指针是一种数据类型
- 指针既然是数据类型，自然可以做函数参数和函数的返回值
- 指针做函数参数的经典例子
  - 两数的互换

# 编写函数实现两数互换

程序 1

程序 2



# 程序1

## 主调函数

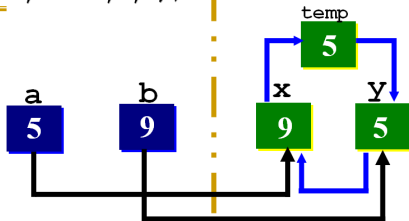
```
main()
{
    int a, b;
    a = 15;
    b = 8;
    Swap(a, b);
    printf("a=%d,b=%d", a,b);
}
```

## 实参

## 形参

## 被调函数

```
void Swap(int x, int y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```



# 程序2

主调函数

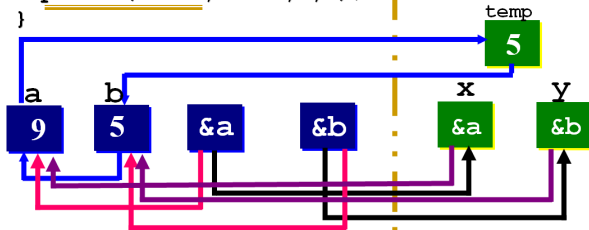
实参

形参

被调函数

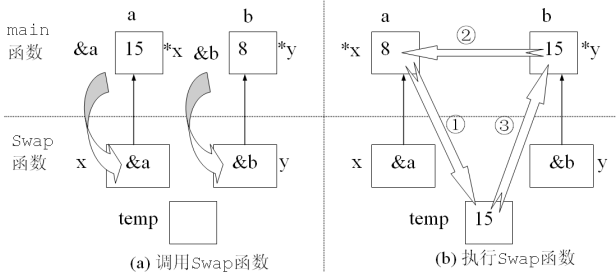
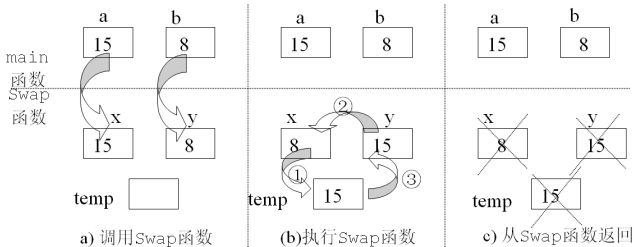
```
main()
{
    int a, b;
    a = 15;
    b = 8;
    Swap(&a, &b);
    printf("a=%d,b=%d", a,b);
}
```

```
void Swap(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
```





# 两种方式比较



# 错误方式1

```
1 void Swap(int *p1, int *p2)
2 {
3     int *p;
4
5     p = p1;    /*p1,p2为内部变量*/
6     p1 = p2;
7     p2 = p;
8 }
```

## 错误方式2

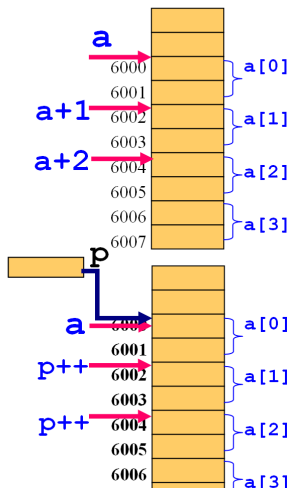
```
1 void Swap(int *p1, int *p2)
2 {
3     int *p;    /*指针p未初始化*/
4
5     *p = *p1;
6     *p1 = *p2;
7     *p2 = *p;
8 }
```

# 函数返回多个值

```
1  GetHighestScore(float *s, int *num,  
2                  char* name)  
3  {  
4      *num = 0001  
5      *s = 100.0f;  
6      strcpy(name, "yan");  
7  }  
8  
9  float score; /*调用 */  
10 int number;  
11 char *name[10];  
12 GetHighestScore(&score, &number, name);
```

# 指针和数组的关系

- 数组名就是一个地址（指针）
  - 只是不能修改这个地址（指针）
  - 可以定义函数的参数为数组
- 指针也可当作数组名使用
  - `int *p, a[10];`
  - `p = a;`
- 数组元素的几种等价引用形式
  - `a[i]`
  - `*(a+i)`



# 下标法

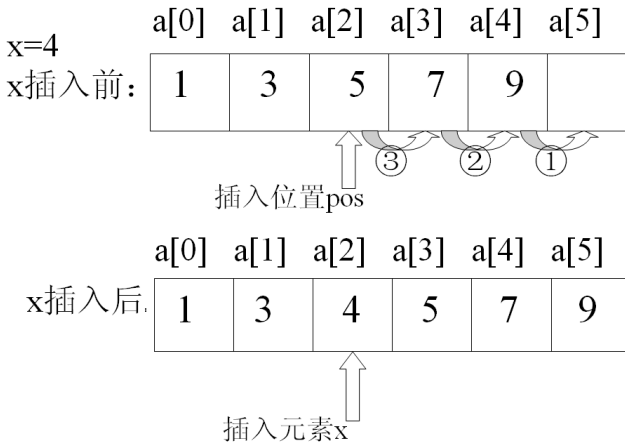
```
1  main()  
2  {  
3      int  a[10];  
4      int  i;  
5  
6      for (i=0; i<10; i++)  
7          scanf("%d", &a[i]);  
8  
9      for (i=0; i<10; i++)  
10         printf("%d ", a[i]);  
11 }
```

# 指针法

```
1  main()  
2  {  
3      int    a[10];  
4      int    *p, i;  
5  
6      for (p=a; p<(a+10); p++)  
7          scanf("%d", p);  
8  
9      for (p=a; p<(a+10); p++)  
10         printf("%d ", *p);  
11 }
```

# 插入排序

- 关键是：找到该插入的位置，然后依次移动插入位置及其后的所有元素腾出这一位置放入待插入的元素





# 插入排序源码

```
1 void Insert(int a[], int n, int x)
2 {
3     int i, pos;
4
5     for (i=0; (i < n) && (x > a[i]); i++)
6     {
7     }
8     pos = i; /*找到插入位置pos*/
9     for (i = n-1; i >= pos; i--)
10    {
11        a[i+1] = a[i]; /*向后移动*/
12    }
13    a[pos] = x; /*插入元素x到位置pos*/
```

# 字符串

- 一串以'\0'结尾的字符。
- C语言并没有为字符串提供任何专门的表示法，完全使用字符数组和字符指针来处理。
  - 字符常量
  - 字符数组
  - 字符指针

```
1 char *p2 = "abc";      /* 字符串常量 */
2 char string[100] = {"abc"}; /* 字符数组 */
3 char *p1 = string;    /* 字符指针 */
```

# 修改字符串

```
1    char *pstr = "hello"; /*修改指针字符串*/
2    pstr[1]='a';          /*这是一个非法操作*/
3
4    char astr[] = "hello"; /*修改数组字符串*/
5    *(astr+1) = 'a';
6
7    printf("%s\n", pstr);
8    printf("%s\n", astr);
```

# 字符指针和字符数组的存储方式

```
1   char *pstr = "hello";  
2   char astr [] = "hello";  
3   printf("hello");
```

## 问题

在你生成的 *exe* 中，到底有多少个 "hello" 拷贝？

# 字符串操作中数组和指针的区别

- 数组名是地址常量
- 字符指针是变量

```
1   char *pstr = "hello"; == const char * p;  
2   /*内容不能更改*/  
3  
4   char astr[] = "hello"; == char const *p ;  
5   /*地址不能更改*/
```

# 问题1

```
1   char *pstr = "hello";
2   char astr []= "hello";
3
4   pstr[1]='a' /*? */
5   pstr=astr; /*? */
6
7   char astr []= "hello";
8   *(astr+1)='a'; /*? */
9   astr = pstr; /*? */
10  astr++;      /*? */
```

# 问题1

```
1   char *pstr = "hello";
2   char astr []= "hello";
3
4   pstr[1]='a' /*? */
5   pstr=astr; /*? */
6
7   char astr []= "hello";
8   *(astr+1)='a'; /*? */
9   astr = pstr; /*? */
10  astr++;      /*? */
```

- 错, 对, 对, 错, 错

## 问题2

```
1   char str[10];  
2   char *ptr;  
3   ptr = "china"           /*? ? */  
4   strcpy(str, "china");  /*? ? */
```



## 问题2

```
1   char str[10];  
2   char *ptr;  
3   ptr = "china"           /*? ? */  
4   strcpy(str, "china");  /*? ? */
```

- 对,错

# 注意事项

- 字符指针变量必须有明确的指向，否则使用是危险的

```
1   char *a;  
2   scanf("%s", a);           /* 错误*/  
3  
4   char str[10];  
5   scanf("%s", str);        /* 正确,但不安全*/  
6   fgets(str,10,stdin);     /* 霸道! */
```

# 总结



# 谢谢大家，欢迎提问！