北京交通大学
BEIJING JIAOTONG UNIVERSITY

函数的参数传递方式

本节内容

- 参数的概念

- 参数的传递方式：
  - 值传递
  - 指针传递
  - 引用传递（C＋＋）

参数的概念

- 实参(argument)：
  - 全称为“实际参数”是
  - 实参可以是常量、
  - 无论实参是何种类
    必须具有确定的值
- 形参(parameter)：
  - 全称为“形式参数”
    虚拟变量。
  - 在定义函数名和函
    接收调用该函数时

```
/*形参的类型在形参表中直接说明*/
int max(int  a,int b)
{
    return (a>b?a:b);
}
int main（）{
  int x=5;
  int y=6;
  printf(“%d”, max(x,y));
}
```

参数的概念

形参与实参的关系

- 形参和实参的功能是作数据传送。
  - 发生函数调用时，主调函数把实参的值传送给被调函数的形参从而实现主调函数向被调函数的数据传送
- 必须注意实参的个数、类型应与形参一一对应，并且实参必须要有确定的值。
- **函数调用中发生的数据传送是单向的。**
  - 即只能把实参的值传送给形参，而不能把形参的值反向地传送给实参。 因此在函数调用过程中，形参的值发生改变，而实参中的值不会变化。

形参与实参的区别

	实参	形参
作用域	实参出现在主调函数中，进入被调函数后，实参变量也不能使用。	形参出现在函数定义中，在整个函数体内都可以使用，离开该函数则不能使用。
内存分配	实参在主调函数中就会被分配内存单元。但在函数调用时，所有主调函数的所有变量（包括存放实参的变量）都会被压栈，在函数内部不起作用。	形参变量只有在被调用时才分配内存单元，在调用结束时，即刻释放所分配的内存单元。

本节内容

- 参数的概念

- 参数的传递方式：
  - 值传递
  - 指针传递
  - 引用传递（C＋＋）

1. 值传递

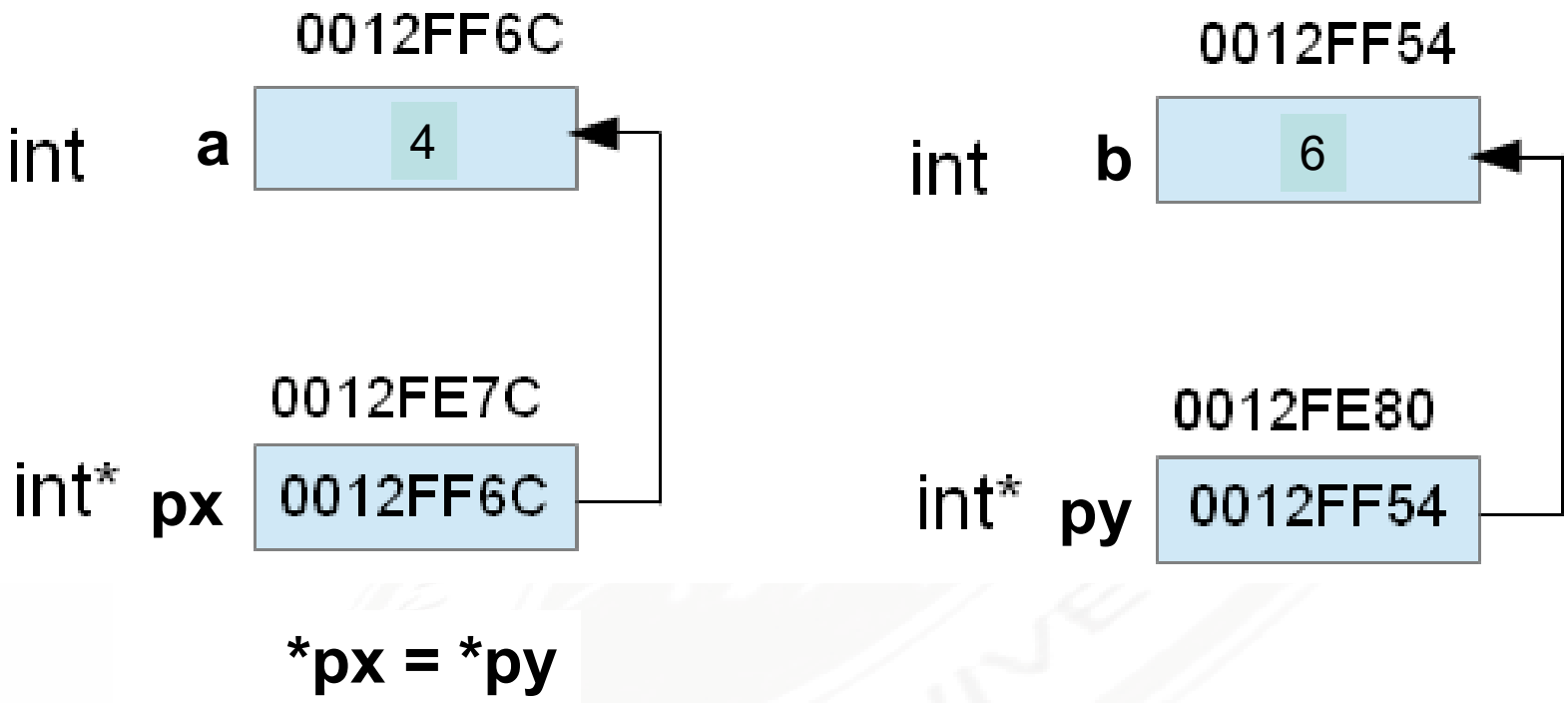
- 参数类型是变量，表达式等。
- 参数的值只能传入，不能传出。
  - 在值传递过程中，
    址中。
  - 形参是实参的拷
    参的值。
- 当函数内部需要修
  影响调用者时，采

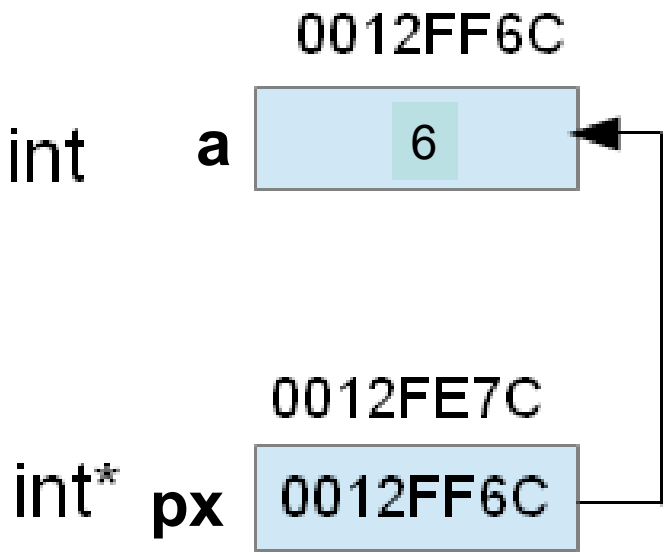
```
1.  void Exchg1(int x, int y)
2.  {
3.      int tmp;
4.      tmp = x;
5.      x = y;
6.      y = tmp;
7.      printf("x = %d, y = %d\n", x, y);
8.  }
9.  main()
10. {
11.     int a = 4,b = 6;
12.     Exchg1(a, b);
13.     printf("a = %d, b = %d\n", a, b);
14.     return(0);
15. }
```

2. 指针传递

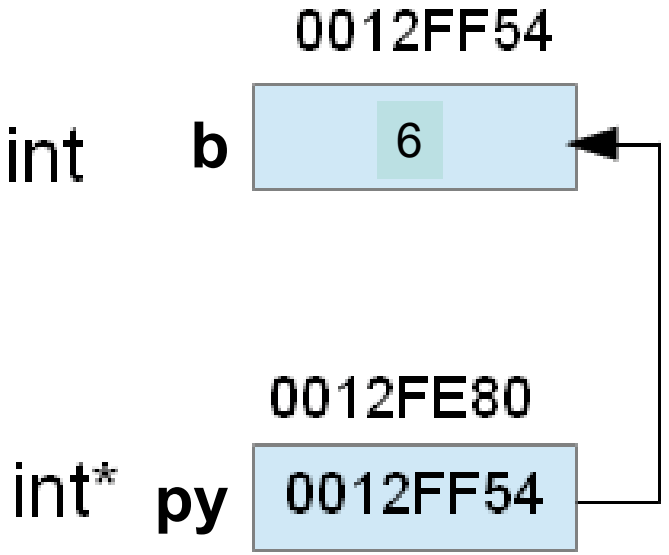
- 形参为指向实参地址的指针。
  - 实参是地址，形参是指针
- 指针传递是一种特殊的值传递。

```
void Exchg2(int *px, int *py)
{
    int tmp = *px;
    *px = *py;
    *py = tmp;
    printf("*px = %d, *py = %d.\n",*px, *py);
}
main()
{
    int a = 4,b = 6;
    Exchg2(&a, &b);
    printf("a = %d, b = %d.\n", a,b);
    return(0);
}
```

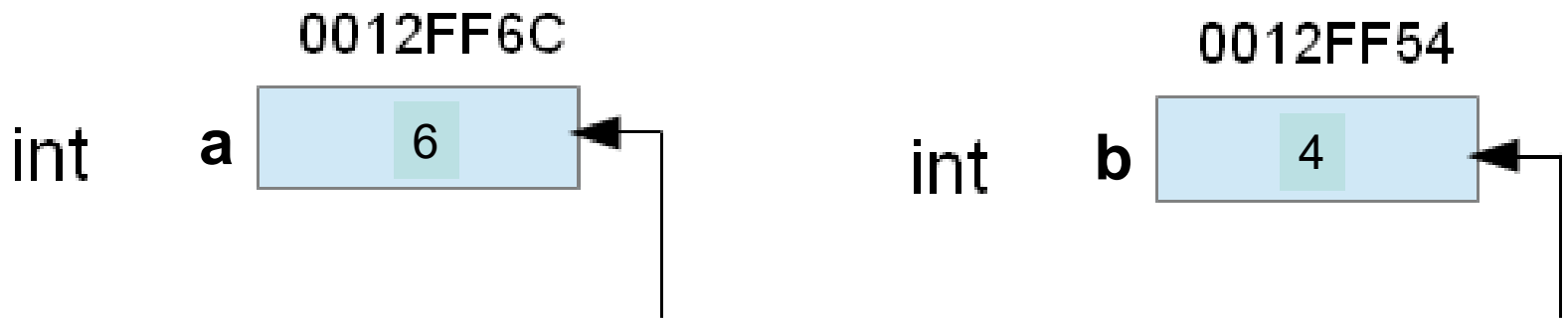

0012FF6C
int
a
4
0012FF54
int
b
6
0012FE7C
int*
px
0012FF6C
0012FE80
int*
py
0012FF54
*px = *py


0012FF6C
int
a
6
0012FE7C
int*
px
0012FF6C

*px = *py;


0012FF54
int
b
6
0012FE80
int*
py
0012FF54

*py = temp;

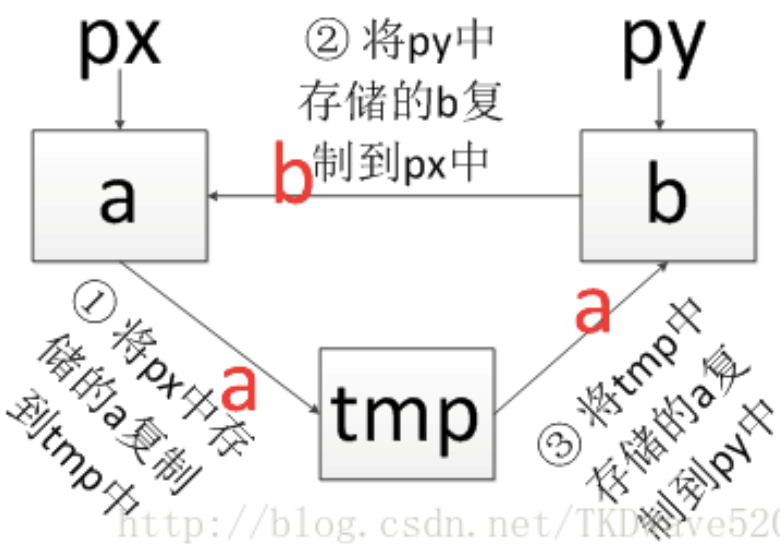


***px = *py;** ***py = temp;**

当对形参的指向操作时，就相当于对实参本身进行的操作。

2. 指针传递

```
void Exchg2(int *px, int *py)
{
    int tmp = *px;
    *px = *py;
    *py = tmp;
    printf("*px = %d, *py = %d.\n",*px, *py);
}
main()
{
    int a = 4,b = 6;
    Exchg2(&a, &b);
    printf("a = %d, b = %d.\n", a,b);
    return(0);
}
```



3. 引用传递（C++）

- 形参相当于是实参的“别名”，对形参的操作其实就是对实参的操作
- 与上两种方式有本质区别
  - 在引用传递过程中，被调函数的形式参数虽然也作为局部变量在栈中开辟了内存空间，但是这时存放的是由主调函数放进来的实参变量的地址。
  - 被调函数对形参的任何操作都被处理成间接寻址，即通过栈中存放的地址访问主调函数中的实参变量。正因为如此，被调函数对形参做的任何操作都影响了主调函数中的实参变量。

```
void Exchg3(int &x, int &y)
{
    int tmp = x;
    x = y;
    y = tmp;
    printf("x= %d,y = %d\n", x, y);
}
main()
{
    int a = 4,b =6;
    Exchg3(a, b);
    printf("a= %d, b = %d\n", a, b);
    return(0);
}
```