



链滴

实验：三种变量交换方法的时间效率对比

作者：[hudk](#)

原文链接：<https://ld246.com/article/1607998858338>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



1、变量的三种交换方式

1.1、借助中间变量

```
long n = 34156236589564L;  
long m = 65485695632451L;  
long temp;  
temp = n;  
n = m;  
m = temp;
```

1.2、使用加减运算

```
long n = 34156236589564L;  
long m = 65485695632451L;  
n = n + m;  
m = n - m;  
n = n - m;
```

1.3、使用异或运算

```
long n = 34156236589564L;  
long m = 65485695632451L;  
n = n ^ m;  
m = n ^ m;  
n = n ^ m;
```

2、测试代码

```

/**
 * 测试三种交换变量的方法各自的运行时间效率
 */
public static void test(){
    //待交换的连个变量
    long n = 34156236589564L;
    long m = 65485695632451L;
    //循环50亿次
    long o = 5000000000L;
    long v = 0;
    //中间变量
    long temp;
    long t1 = System.currentTimeMillis();
    //【第一种方式】
    //借用中间变量交换
    //这种变量交换的方式语义上相对好理解
    //通过测试，运行效率在这三种当中相对最好
    while (v < o){
        temp = n;
        n = m;
        m = temp;
        v++;
    }
    v = 0;
    long t2 = System.currentTimeMillis();
    //【第二种方式】
    //用加减法变量交换
    //这种方式容易产生溢出
    //通过测试，运行效率相对第一种较差
    while (v < o){
        n = n + m;
        m = n - m;
        n = n - m;
        v++;
    }
    v = 0;
    long t3 = System.currentTimeMillis();
    //【第三种方式】
    //借用异或变量交换
    //使用位运算来实现变量的交换
    //通过测试，运行效率相对第一种较差，相对第二种几乎一样
    while (v < o){
        n = n ^ m;
        m = n ^ m;
        n = n ^ m;
        v++;
    }
    long t4 = System.currentTimeMillis();
    System.out.println("借用中间变量交换所花时间: " + (t2 - t1));
    System.out.println("用加减法变量交换所花时间: " + (t3 - t2));
    System.out.println("借用异或变量交换所花时间: " + (t4 - t3));
}

public static void main(String[] args) {

```

```
test();  
}
```

3、运行结果(毫秒)

借用中间变量交换所花时间: 1633

用加减法变量交换所花时间: 3237

借用异或变量交换所花时间: 3226

4、结论

这三种方式的运行时间效率: 第一种方式 > 第二种方式, 第二种和第三种效率相当。

在实验之前, 一般直觉, 认为位运算的效率会更高, 但是通过实验表明, 通过中间变量的方式, 运算率是最高的。

这是为什么呢? 目前还不清楚, 有待进一步思考和理解。。。