

• [首页](#) • [开源项目](#) • [问答](#) • [代码](#) • [博客](#) • [翻译](#) • [资讯](#) • [移动开发](#) • [招聘](#) • [城市圈](#)
当前访客身份：游客 [[登录](#) | [加入开源中国](#)]

在 39900 款开源软件中

软件

软件

搜索



[jiachcheng](#) ♂ [关注此人](#)

[关注\(0\)](#) [粉丝\(2\)](#) [积分\(4\)](#)



这个人很懒，啥也没写

[发送私信](#) [请教问题](#)

博客分类

- [工作日志](#)(4)
- [日常记录](#)(0)
- [转贴的文章](#)(0)

阅读排行

1. [1. Java 中int与byte数组转换详解](#)
2. [2. Java socket ,多客户端同时与服务端不停的交换数据](#)
3. [3. 关于ThreadPoolExecutor 调用RejectedExecutionHandler的机制](#)
4. [4. ORA-01688:扩展ORACLE表空间操作](#)

最新评论

- [@会打杂的前端攻城狮](#)：你好。我想问一下 while(true){ 056 try { ... [查看»](#)
- [@taxuezhuifeng](#)：测试一下，可以正常运行，但是不知道为什么可以，... [查看»](#)

访客统计

- 今日访问：0
- 昨日访问：14
- 本周访问：0
- 本月访问：628
- 所有访问：26684

[空间](#) » [博客](#) » [工作日志](#)

原 Java 中int与byte数组转换详解

发表于3年前(2012-12-21 15:42) 阅读 (16950) | 评论 (0) 22人收藏此文章, [我要收藏](#)
赞2

1、与运算符的理解 (&) :

参加运算的两个数据，按二进位进行“与”运算。如果两个相应的二进位都为1，则该位的结果值为1，否则为0。即 $0 \& 0 = 0$; $0 \& 1 = 0$; $1 \& 0 = 0$; $1 \& 1 = 1$;

例子 : `int x = 62255550; y = 0xff;`

转二进制 : `x = 62255550 = 11101101011111000110111110;`

`y = 0xFF = 11111111;`

11 10110101 11110001 10111110

`&00 00000000 00000000 11111111`

00 00000000 00000000 10111110

(从右到左做与运算，不足的补0)

如果参加&运算的是负数(如-3 & -5)，则以补码形式表示为二进制数，然后按位进行“与”运算。

2、移位运算 : java移位运算符三种 : << (左移)、>> (带符号右移) 和 >>> (无符号右移)。

--1、左移运算符

左移运算符<<使指定值的所有位都左移规定的次数。

1) 它的通用格式如下所示 :

`value << num`

num 指定要移位值value 移动的位数。

左移的规则只记住一点：丢弃最高位，0补最低位

如果移动的位数超过了该类型的最大位数，那么编译器会对移动的位数取模。如对int型移动33位，



实际上只移动了 $33\%32=1$ 位。

2) 运算规则

按二进制形式把所有的数字向左移动对应的位数，高位移出(舍弃)，低位的空位补零。

当左移的运算数是int 类型时，每移动1位它的第31位就要被移出并且丢弃；

当左移的运算数是long 类型时，每移动1位它的第63位就要被移出并且丢弃。

当左移的运算数是byte 和short类型时，将自动把这些类型扩大为 int 型。

3) 数学意义

在数字没有溢出的前提下，对于正数和负数，左移一位都相当于乘以2的1次方，左移n位就相当于乘以2的n次方

4) 计算过程：

例如：3 <<2(3为int型)

1) 把3转换为二进制数字0000 0000 0000 0000 0000 0000 0000 0011，

2) 把该数字高位(左侧)的两个零移出，其他的数字都朝左平移2位，

3) 在低位(右侧)的两个空位补零。则得到的最终结果是0000 0000 0000 0000 0000 0000 0000 1100，

转换为十进制是12。

--2、右移运算符

右移运算符<<使指定值的所有位都右移规定的次数。

1) 它的通用格式如下所示：

value >> num

num 指定要移位值value 移动的位数。

右移的规则只记住一点：符号位不变，左边补上符号位



2) 运算规则：

按二进制形式把所有的数字向右移动对应的位数，低位移出(舍弃)，高位的空位补符号位，即正数补零，负数补1

当右移的运算数是byte 和short类型时，将自动把这些类型扩大为 int 型。

例如，如果要移走的值为负数，每一次右移都在左边补1，如果要移走的值为正数，每一次右移都在左边补0，这叫做符号位扩展（保留符号位）（sign extension），在进行右移操作时用来保持负数的符号。

3) 数学意义

右移一位相当于除2，右移n位相当于除以2的n次方。

4) 计算过程

11 >> 2(11为int型)

1)11的二进制形式为：0000 0000 0000 0000 0000 0000 0000 1011

2)把低位的最后两个数字移出，因为该数字是正数，所以在高位补零。

3)最终结果是0000 0000 0000 0000 0000 0000 0000 0010。

转换为十进制是2。

35 >> 2(35为int型)

35转换为二进制：0000 0000 0000 0000 0000 0000 0010 0011

把低位的最后两个数字移出：0000 0000 0000 0000 0000 0000 0000 1000

转换为十进制：8

5) 在右移时不保留符号位的出来

右移后的值与0x0f进行按位与运算，这样可以舍弃任何的符号位扩展，以便得到的值可以作为定义数组的下标，从而得到对应数组元素代表的十六进制字符。

--3、无符号右移

无符号右移运算符>>>

它的通用格式如下所示：

value >>> num

num 指定要移位值value 移动的位数。

无符号右移的规则只记住一点：忽略了符号位扩展，0补最高位

无符号右移运算符>>> 只是对32位和64位的值有意义



3、java中：

1	一个byte 占一个字节，即8位比特；
2	一个int 占4个字节，即32比特；
3	<p>java的二进制采用的是补码形式</p> <p>(1)一个数为正，则它的原码、反码、补码相同</p> <p>(2)一个数为负，符号位为1，其余各位是对原码取反，然后整个数加1</p> <p>因为补码存在，所以右移运算后要与0xff相与运算</p>
4	在Java中，当我们要将int 转换为byte数组时，一个int就需要长度为4个字节的数组来存放，其中一次从数组下标为[0]开始存放int的高位到低位。
5	<p>Java中的一个byte，其范围是-128~127的，而Integer.toHexString的参数本来是int，如果不进行&0xff，那么当一个byte会转换成int时，对于负数，会做位扩展，举例来说，一个byte的-1（即0xff），会被转换成int的-1（即0xffffffff），那么转化出的结果就不是我们想要的了。</p> <p>而0xff默认是整形，所以，一个byte跟0xff相与会先将那个byte转化成整形运算，这样，结果中的高的24个比特就总会被清0，于是结果总是我们想要的。</p>

4、代码：

```
1  /**
2      * int到byte[]
3      * @param i
4      * @return
5      */
6  public static byte[] intToByteArray(int i) {
7      byte[] result = new byte[4];
8      //由高位到低位
9      result[0] = (byte)((i >> 24) & 0xFF);
10     result[1] = (byte)((i >> 16) & 0xFF);
11     result[2] = (byte)((i >> 8) & 0xFF);
12     result[3] = (byte)(i & 0xFF);
```



```

13         return result;
14     }
15
16     /**
17     * byte[]转int
18     * @param bytes
19     * @return
20     */
21     public static int byteArrayToInt(byte[] bytes) {
22         int value= 0;
23         //由高位到低位
24         for (int i = 0; i < 4; i++) {
25             int shift= (4 - 1 - i) * 8;
26             value +=(bytes[i] & 0x000000FF) << shift;//往高位游
27         }
28         return value;
29     }

```

分享到： 新浪微博 weibo.com  腾讯微博 t.qq.com 赞 2

声明：OSCHINA 博客文章版权属于作者，受法律保护。未经作者同意不得转载。

- [« 上一篇](#)
- [下一篇 »](#)

最新热门职位

更多开发者职位上 [开源中国·招聘](#)



java开发工程师（集团

客... 远传技术

月薪：5-10K
数据挖掘 微播易

月薪：15-30K



自然语言处理工程师 美

柚

月薪：10-20K
技术总监 北京

月薪：25-35K



评论0



插入：[表情](#) [开源软件](#)

[关闭](#)插入表情

发表评论

关闭相关文章阅读

- 2015/07/28 [Java中int 转 byte](#)
- 2015/10/13 [为什么InputStream.read\(\)读取一个b...](#)
- 2014/11/06 [Java enum vs int](#)
- 2013/05/13 [java中Integer和int](#)
- 2015/04/29 [java中char和byte的转换 ...](#)

© 开源中国(OSChina.NET) | [关于我们](#) | [广告联系](#) | [@新浪微博](#) | 开源中国手机客户端：
[开源中国手机版](#) | 粤ICP备12009483号-3

开源中国社区(OSChina.net)是工信部 [开源软件推进联盟](#) 指定的官方社区

