

简单信息的表达和运算

位运算

赵英良



1.什么是位运算？

- ▶ 数据在计算机中是以二进制数的形式存放的。

a=0110 1001B

b=0101 1110B

	最高位					最低位			
	7	6	5	4	3	2	1	0	
a	0	1	1	0	1	0	0	1	
b	0	1	0	1	1	1	1	0	

位运算只能对整数进行

2.位运算有哪些？

- ▶ 与(and)、或(or)、取反(not)、异或(xor)、左移、右移
- ▶ 按位“与”、按位“或”、按位“取反”、按位“异或”

按位“与”

ai	0	0	1	1
bi	0	1	0	1
结果	0	0	0	1

按位“或”

ai	0	0	1	1
bi	0	1	0	1
结果	0	1	1	1

按位“异或”

ai	0	0	1	1
bi	0	1	0	1
结果	0	1	1	0

按位“取反”

ai	0	1
结果	1	0

移位运算

移位：将二进制数的每一位 向左 或 向右 移动 指定的数位

如：左移两位

如：右移两位

3.位运算符

- ▶ & 按位 “与”
- ▶ | 按位 “或”
- ▶ ^ 按位 “异或” , Shift+<6>
- ▶ << 左移
- ▶ >> 右移
- ▶ ~ 按位 “取反” , Shift+<1左边的符号>
- ▶ 格式
 <整数类型表达式1> <位运算符> <整数类型表达式2>
 ~<整数类型表达式>

4. 举例

a=0110 1001B

105

69H

0X69

b=0101 1110B

94

5EH

0X5E

a=0110 1001B

& b=0101 1110B

c=0100 1000B

72

0X48

4. 举例

a=0110 1001B	105	69H	0X69
b=0101 1110B	94	5EH	0X5E
c=a&b;	72		0x48

a=0110 1001B		
b=0101 1110B		
<hr/>		
c=0111 1111B	127	0X7F

4. 举例

a=0110 1001B	105	69H	0X69
b=0101 1110B	94	5EH	0X5E
c=a&b;	72		0x48
c=a b;	127		0x7f

a=0110 1001B		
^ b=0101 1110B		
<hr/>		
c=0011 0111B	55	0X37

4. 举例

为什么这里a写32位?

a=0110 1001B	105	69H	0X69
b=0101 1110B	94	5EH	0X5E
c=a&b;	72		0x48
c=a b;	127		0x7f
c=a^b;	55		0x37

a=00000000 00000000 00000000 0110 1001B

~a

c=~a=11111111 11111111 11111111 1001 0110B

c: -106

0xFFFFFFFF96

4. 举例

a=0110 1001B	105	69H	0X69
b=0101 1110B	94	5EH	0X5E
c=a&b;	72		0x48
c=a b;	127		0x7f
c=a^b;	55		0x37
c=~a;	-106		0xFFFFFFFF96

a 00000000 00000000 00000000 0110 1001B<<2

c= 00B

c: 420

0X1A4

4. 举例

a=0110 1001B	105	69H	0X69
b=0101 1110B	94	5EH	0X5E
c=a&b;	72		0x48
c=a b;	127		0x7f
c=a^b;	55		0x37
c=~a;	-106		0xFFFFFFFF96
c=a<<2;	420		0x1A4
a 00000000 00000000 00000000 0110 1001B>>2			
c=00			B
c: 26			0X1A

4. 举例

a=0110 1001B	105	69H	0X69
b=0101 1110B	94	5EH	0X5E
c=a&b;	72		0x48
c=a b;	127		0x7f
c=a^b;	55		0x37
c=~a;	-106		0xFFFFFFFF96
c=a<<2;	420		0x1A4
c=a>>2;	26		0x1A

看完整程序

5.位运算符的优先级

高

按位反 逻辑非! , 负号- , 前置++ , --

算术 乘、除、求余 * / %

算术加、减 + -

按位左移<<、右移>>

从左向右

关系: 大于小于 >,>=,<,<=

关系: 等于不等于 ==, !=

按位与

按位异或

从左向右

按位或

逻辑 与 &&

逻辑或 ||

...

等号=

从右向左

圆括号

低

6.位运算的应用

按位"与"可以将某些位设置为0

a=0110 1001B

& b=1111 0000B

c =0110 0000B

按位"或"可以将某些位设置为1

a=0110 1001B

| b=1111 0000B

c=1111 1001B

位运算的应用

$$1 \wedge 1 \rightarrow 0$$

$$0 \wedge 1 \rightarrow 1$$

和1"异或"变反

$$1 \wedge 0 \rightarrow 1$$

$$0 \wedge 0 \rightarrow 0$$

和0"异或"不变

按位"异或"可以让某些位变反，而另一些位不变

a=0110 1001B

\wedge b=1111 0000B

c=1001 1001B

7. 复合位运算

▶ 位运算可以使用复合运算符

- 复合与: $\&=$ 复合或: $|=$
- 复合异或: $\wedge=$
- 复合左移: $\ll=$ 复合右移: $\gg=$

▶ `int x=0x1A6F,y=0x35AA,k=3:`

- `x&y=;` 等价于 `x=x&y;`
- `x|y=;` 等价于 `x=x|y;`
- `x^y=;` 等价于 `x=x^y;`
- `x<<=k;` 等价于 `x=x<<k;`
- `x>>=k;` 等价于 `x=x>>k;`

优先级：很低
与复合算术运算、等号同级
结合顺序：从右向左

从右向左：
! ~ + - 前置++ 前置--

8.位运算使用注意事项

- ▶ (1)一个&和两个&&是不同的

&按位“与” &&逻辑“与”

|按位“或” ||逻辑“或”

- (2)在输出中使用移位运算的表达式时，要加括号

```
cout<<4<<2<<endl; //结果42
```

```
cout<<(4<<2)<<endl; //结果16
```

重载

总结

▶ 位运算

- 对整数的二进制位进行的操作
- 与 $\&$ 、 或 $|$ 、 异或 \wedge 、 取反 \sim 、 左移 \ll 、 右移 \gg

▶ 优先级

- \sim > 算术运算 > 比较运算 > $\&$ > \wedge > $|$

▶ 位运算只能作用于整型数据

- 操作数可以是变量、常量或表达式