

第三章 运算符与表达式

主讲：刘晓光

张海威 张莹

殷爱茹 李雨森

宋春瑶 沈玮

卢少平



南开大学

计算机学院&网络空间安全学院



逻辑运算 ☐ 位运算符
位运算 ☒ 位运算表达式求值
条件运算 ☐ 优先级与结合性
其它运算 ☐ 位运算中的类型转换



基本概念



赋值运算



算术运算



关系运算



逻辑运算



位运算



条件运算



其它运算



位运算的基本概念

位运算是一种对运算分量按二进制位进行操作的运算，而且是**作用于运算分量的每一个二进制位上**（并不是只对其中的某一个位进行运算）。位运算的**运算对象只能是整型数据**（包括字符型），且运算结果仍为整型数据



位运算符 (Bitwise operators)

单目按位取反

- \sim

双目运算符

- 按位与 $\&$
- 按位或 $|$
- 按位异或 \wedge
- 按位左移 \ll
- 按位右移 \gg



位运算表达式求值

运算步骤

- 将运算分量用相应位数的二进制表示
- 将运算分量的二进制形式按位进行相应的运算
- 运算结果需根据实际情况补0

位运算表达式的值

- 根据运算符的含义进行位运算得到的结果
 - 二进制形式
- 运算结果仍然以十进制的形式输出



【例】

unsigned char d1=38, d2=44;

- 设有无符号字符型变量d1，它具有初值38，即二进制的00100110；及无符号字符型变量d2，它具有初值44，即二进制的00101100。
- d1&d2的结果将是36(由00100110与00101100进行按位与，即逐位进行与运算，得结果00100100，即10进制的36)；
- d1|d2的结果将是46(由00100110与00101100进行按位或，即逐位进行或运算，得结果00101110)；



unsigned char d1=38, d2=44;

- $d1 \wedge d2$ 的结果将是 10 (由 00100110 与 00101100 进行按位异或, 即 **逐位进行异或** 运算, 得结果 00001010);
- $d1 \ll 1$ 的结果将是 76 (将 00100110 的 **每一位都向左移动** 一个位后得结果 01001100)

0

0	1	0	0	1	1	0	0
---	---	---	---	---	---	---	---



unsigned char d1=38, d2=44;

- **d1>>1**的结果将是19(将00100110的**每一位**都**向右移动**一个位后得结果00010011)

0	0	0	1	0	0	1	1
---	---	---	---	---	---	---	---

0

- **将~d1的结果赋值给无符号字符型变量d3, 而后按int值输出d3将得结果217(将00100110的**每一位**都**取反**后得结果11011001)**




```
#include <iostream.h>
void main() {
    unsigned char d1=38, d2=44;
    cout<<"unsigned char d1=38, d2=44;"<<endl;
    → cout<<"d1="<<d1<<"          d2="<<d2<<endl;
    → cout<<"int (d1)="<<int (d1)<<"int (d2)="<<int (d2)<<endl;
    cout<<"-----"<<endl;
    cout<<"d1&d2  => " << (d1&d2)<<endl;
    cout<<"d1|d2  => " << (d1|d2)<<endl;
    cout<<"d1^d2  => " << (d1^d2)<<endl;
    cout<<"d1<<1  => " << (d1<<1)<<endl;
    cout<<"d1>>1  => " << (d1>>1)<<endl;
    cout<<"~d1    => " << (~d1)<<endl;
    unsigned char d3 = ~d1;
    cout<<"-----"<<endl;
    cout<<"unsigned char d3=~d1;"<<endl;
    → cout<<"int (d3)  => " << int (d3)<<endl;
    cout<<"-----"<<endl;
}
```



运行结果

```
unsigned char d1=38, d2=44;
d1=&          d2=,
int(d1)=38    int(d2)=44
```

```
-----
d1&d2 => 36
d1|d2 => 46
d1^d2 => 10
d1<<1 => 76
d1>>1 => 19
~d1    => -39
```

```
-----
unsigned char d3=~d1;
int(d3) => 217
-----
```

