

Богатов Р.Н.

Программирование на языке высокого уровня

Лекция 15.
Битовые операции

Кафедра АСОИУ ОмГТУ, 2013

«Логические» vs. «битовые»

- Двоичные логические операции

- отрицание (\neg)
- конъюнкция (\wedge или $\&$)
- дизъюнкция (\vee)
- сложение по модулю два (\oplus)

Конъюнкция

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

Дизъюнкция

a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

Сложение по модулю 2

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

- Битовые операции

- побитовое отрицание
- побитовое И
- побитовое ИЛИ
- сложение по модулю два
- циклический сдвиг

and	01000011	xor	01000011
	01110010		01110010
	01000010		00110001
or	01000011	not	01000011
	01110010		10111100
	01110011		

Применение битовых операций

- Проверка битов:

$$\begin{array}{r} \text{and} \quad 00100110 \\ \quad \quad 11111111 \\ \hline \quad \quad 00100110 \end{array}$$

- Обнуление битов:

$$\begin{array}{r} \text{and} \quad 00100110 \\ \quad \quad 11111011 \\ \hline \quad \quad 00100010 \end{array}$$

- Установка битов в единицу:

$$\begin{array}{r} \text{or} \quad 00100110 \\ \quad \quad 00000000 \\ \hline \quad \quad 00100110 \end{array}$$

$$\begin{array}{r} \text{or} \quad 00100110 \\ \quad \quad 00001000 \\ \hline \quad \quad 00101110 \end{array}$$

- Смена значений битов:

$$\begin{array}{r} \text{xor} \quad 00100110 \\ \quad \quad 00000010 \\ \hline \quad \quad 00100100 \end{array}$$

$$\begin{array}{r} \text{xor} \quad 00100110 \\ \quad \quad 11111111 \\ \hline \quad \quad 11011001 \end{array}$$

- Операции побитового циклического сдвига.

$$01101001 \text{ shl } 3 = 01001000$$

$$01101001 \text{ shr } 4 = 00000110$$

Языки программирования

- Логические операции

Язык	НЕ	И	ИЛИ	Искл. ИЛИ	Эквив.	Не экв.
C++ ^[2]	!	&&		^	==	!=
Pascal ^[5]	not	and	or	xor	=	<>

```
if (a != 0 && b/a > 3)
{
    ...
}
```

- Побитовые операции

Язык	НЕ	И	ИЛИ	Искл. ИЛИ	Сдвиг влево	Сдвиг вправо
C/C++, Java, C#, Ruby ^[4]	~	&		^	<<	>>
Pascal ^[5]	not	and	or	xor	shl	shr

Побитовые чудеса...

```
// Подсчёт ненулевых
uint CountBits(uint x)
{
    uint c = 0;
    for (; x != 0; x >> 1)
        c += x & 1;

    return c;
}
```

```
// Подсчёт ненулевых
uint CountBits(uint x)
{
    uint c = 0;
    for (; x != 0; c++)
        x &= x - 1;
    return c;
}
```

```
// Подсчёт
uint Count
{
    x = x >> 16;
    x = x >> 8;
    return
}
```

```
// Вычисление бита чётности
bool ParityBit(uint x)
{
    return (CountBits(x) % 2 != 0);
}
```

```
// Вычисление бита чётности. Оптимизация ;- )
bool ParityBit(uint x)
{
    bool parity = false;
    while (x != 0)
    {
        parity = !parity;
        x = x & (x - 1);
    }
    return parity;
}
```

```
// Вычисление бита чётности. Эммм... 8- /
bool ParityBit(uint x)
{
    x ^= x >> 16;
    x ^= x >> 8;
    x ^= x >> 4;
    x &= 0xf;
    return ((0x6996 >> (int)x) & 1) > 0;
}
```

Больше «чудес»
на странице
[bithacks.html](#)
(так её и ищите)

Сложение по модулю 2

OR

eXclusive

Свойства

- $a \oplus 0 = a$
- $a \oplus a = 0$
- $a \oplus b = b \oplus a$
- $a \oplus 1 = \bar{a}$
- $(a \oplus b) \oplus b = a$
- $\bar{a} \oplus b = a \oplus \bar{b}$

X	Y	Z	$\oplus(X,Y,Z)$
0	0	0	0
1	0	0	1
0	1	0	1
1	1	0	0
0	0	1	1
1	0	1	0
0	1	1	0
1	1	1	1

Пригодается:

- инверсия по маске
- контроль чётности
- обмен значений переменных
- спрайтовая графика
- криптография

```
// обмен переменных
...
int t = x;
x = y;
y = t;
...
```

```
// обмен переменных
// без посредника!!!
...
x = x^y;
y = x^y;
x = x^y;
...
```

Шифрование файла

```
{  
    FileStream f1 = new FileStream(textBox1.Text, FileMode.Open);  
    FileStream f2 = new FileStream(textBox2.Text, FileMode.Create);  
    int L = progressBar1.Maximum = (int)f1.Length;  
  
    string pwd = textBox3.Text;  
    for (int i = 0; i < L; i++)  
    {  
        byte x = (byte)f1.ReadByte();  
        byte p = (byte)pwd[i % pwd.Length];  
        byte y = (byte)(x ^ p);  
        f2.WriteByte(y);  
        progressBar1.Value = i;  
    }  
    f1.Close();  
    f2.Close();  
}
```