

Внутреннее представление данных

Юрий Литвинов
yurii.litvinov@gmail.com

05.10.2018

Побитовые операции

- ▶ `&` — побитовое “И”
- ▶ `|` — побитовое “ИЛИ”
- ▶ `~` — побитовое “НЕ”
- ▶ `1 & 2 == false`, но `1 && 2 == true`
- ▶ `<<`, `>>` — битовый сдвиг
 - ▶ `int x = 1 << 3`
- ▶ `sizeof` — размер типа в байтах
 - ▶ `int s = sizeof(int) * 8`
- ▶ Обратите внимание, что ВСЁ хранится как набор бит
 - ▶ “3” — литерал, лишь удобная форма записи 00...0011 в коде

Маски

`&`:

1	1	0	1	1	0	1	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0

`&`:

1	1	0	1	1	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0

Работа с масками

```
char x = 5;
```

```
int bit = 0b10000000;
```

```
for (int j = 0; j < 8; ++j)
```

```
{
```

```
    printf((x & bit) ? "1" : "0");
```

```
    bit = bit >> 1;
```

```
}
```

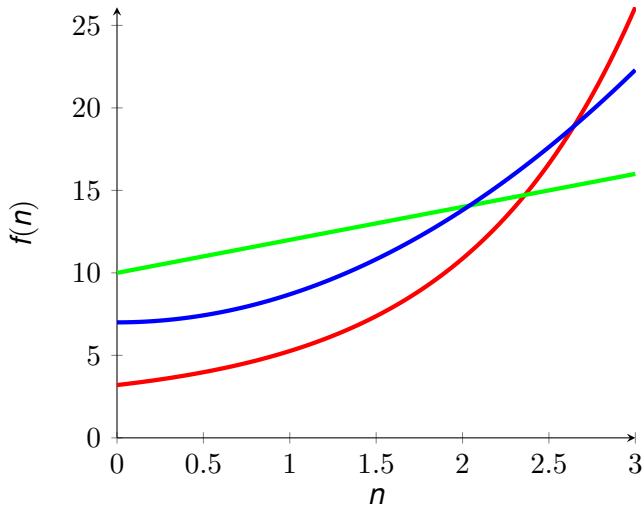
Целые числа

- ▶ Прямой код
 - ▶ 5 — 00000101, -5 — 10000101
- ▶ Обратный код
 - ▶ 5 — 00000101, -5 — 11111010
- ▶ Дополнительный код
 - ▶ 5 — 00000101, -5 — 11111011
 - ▶ $-x$ представляется как $2^n - x$, поэтому и дополнительный
 - ▶ n — разрядность регистра

Арифметические действия

- ▶ В обратном коде единица переноса в старшем разряде прибавляется к младшему разряду
- ▶ В дополнительном коде единица переноса в старшем разряде отбрасывается

OLOLO



TROLOLO

1	1	0	1	1	0	1	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0

1	1	0	1	1	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0