

## pg22 浮点运算不可结合

由于表示的精度有限，浮点运算是不可结合的

e.g:  $(3.14+1e20)-1e20$ 的值为0.0, 而 $3.14+(1e20-1e20)$ 的值是3.14

## pg24 GCC

GCC (GNU Compiler Collection) : GNU编译器套装

## pg27 32位程序/64位程序的区别

对于一个字长为 $w$ 位的机器而言，虚拟地址的范围为 $0 \sim 2^w - 1$ ，程序最多访问 $2^w$ 个字节

将程序称为“32位程序”或者“64位程序”时，区别在于该程序是如何编译的，而不是其运行的机器类型

## 布尔运算

- pg36  $|$  和  $\&$  有结合律、分配律

$$a \& (b | c) = (a \& b) | (a \& c)$$

$$a | (b \& c) = (a | b) \& (a | c)$$

- pg36 异或

$$a \wedge 0 = a$$

$$a \wedge a = 0$$

$$a \wedge \text{全1} \text{ 等价于 } \sim a$$

- 练习题2.12-B 位运算实现  $x==y$  :  $!(x \wedge y)$
- 练习题2.10 位运算实现原地交换

```
1 void inplace_swap(int *x, int *y) {
2     *y = *x ^ *y;
3     *x = *x ^ *y;
4     *y = *x ^ *y;
5 }
```

## pg48 补码、反码

- 补码 (*Two's complement*) : 对于非负数 $x$ ，我们用 $2^w - x$  (这里只有一个1) 表示 $-x$
- 反码 (*Ones' complement*) : 对于非负数 $x$ ，我们用 $[111...1] - x$  (这里有很多个1) 表示 $-x$

## 溢出判断

- 练习题2.27 无符号加法判断溢出

```
1  int uadd_ok(unsigned x, unsigned y) {
2      unsigned sum = x+y;
3      return sum >= x;
4  }
```

- 练习题2.30 补码加法判断溢出

```
1  int tadd_ok(int x, int y) {
2      int sum = x+y;
3      int neg_over = x < 0 && y < 0 && x+y >= 0;
4      int pos_over = x >= 0 && y >= 0 && x+y < 0;
5      return !neg_over && !pos_over;
6  }
```

- 练习题2.35 补码乘法判断溢出

```
1  int tmul_ok(int x, int y) {
2      int p = x*y;
3      return !x || p/x == y;
4  }
```

## 乘以常数优化

TODO