

Технология программирования на ЭВМ. Числа и арифметика.

Баев А.Ж.

Казахстанский филиал МГУ

19 октября 2019

Виды целых чисел

Целые числа различаются:

1. по размеру (сколько байт занимает число: 1, 2, 4 или 8);
2. беззнаковые (10, 20) и знаковые (+10, -20).

Беззнаковые числа

Стандартные размеры: 1, 2, 4 или 8 байта (то есть 8, 16, 32 или 64 бита соответственно).

Так как каждый бит может принимать значения 1 или 0, то общее количество кодируемых вариантов будет 2^{BIT} .

type	byte	min	max
unsigned char	1	0	$2^8 - 1$ (255)
unsigned short	2	0	$2^{16} - 1$ (65 535)
unsigned int	4	0	$2^{32} - 1$ (4 294 967 295)
unsigned long long	8	0	$2^{64} - 1$ (18 446 744 073 709 551 615)

Беззнаковые типы, как правило, используются для подсчета количества, длины, размера.

Знаковые числа

Стандартные размеры: 1, 2, 4 или 8 байта (то есть 8, 16, 32 или 64 бита соответственно).

Так как каждый бит может принимать значения 1 или 0, то общее количество кодируемых вариантов будет 2^{BIT} .

type	byte	min	max
char	1	-2^7 (-128)	$2^7 - 1$ (127)
short	2	-2^{15} ($-32\,768$)	$2^{15} - 1$ ($32\,767$)
int	4	-2^{31} ($-2\,147\,483\,648$)	$2^{31} - 1$ ($\approx 2 \cdot 10^9$)
long long	8	-2^{63} ($-9\,223\,372\,036\,854\,775\,808$)	$2^{63} - 1$ ($\approx 9 \cdot 10^{18}$)

Знаковые типы, как правило, используются для всего остального.

Константы

- 42 — тип `int`;
- 42U — тип `unsigned int`;
- 42LL — тип `long long`;
- 42ULL — тип `unsigned long long`;
- 052 — тип `int` в восьмиричной системе счисления;
- 0x2A — тип `int` в шестнадцатиричной системе счисления.

Префиксы 0x и 0 задают 16-ричную и 8-ричную систему счисления соответственно.

Суффикс U обозначает беззнаковый тип.

Суффикс LL обозначает тип `long long`.

Вещественные чисел в компьютере нет.

Так как вещественных чисел на любом конечном отрезке — несчетно, то хранить все числа из любого, даже небольшого диапазона в конечных ресурсах компьютера невозможно. Но можно хранить рациональное приближение!

Числа с плавающей запятой.

Запишем число 2,5 в двоичной системе счисления:

$$2,5 = 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} = 10,1_2$$

Нормализуем его:

$$10,1_2 = 1.01_2 \cdot 2^1$$

Числа с плавающей запятой.

Запишем число 0,1 в двоичной системе счисления (умножаем на 2 и смотрим целую часть):

$$0,1 = 0,00011001100110011.._2 = 0,0(0011)_2$$

Нормализуем его (получим «экспоненту»):

$$0,1 = 0,00011001100110011.._2 = 1,1001100110011.._2 \cdot 2^{-4}$$

Оставим только 8 бит после запятой (получим «мантиссу»):

$$0,1 \approx 1,10011001 \cdot 2^{-4}$$

Мантисса: $m = 10011001$

Экспонента: $= -4$

Числа с плавающей запятой.

Общий вид

$$\pm(1 + m) \cdot 2^e.$$

Числа около нуля идут достаточно плотно с шагом значительно меньше 1, число близкие к максимальным значениям идут шагом значительно больше 1.

Наиболее плотно числа расположены возле ± 1.0 .

Числа с плавающей запятой.

Наблюдение: двоично-рациональные числа представляются точно, а любые остальные нецелые числа — приближенно (то есть, 0.25 хранится точно, а 0.1 — приближенно).

Типы чисел.

тип	размер	мантисса	экспонента	диапазон целых
float	4 байта	23 бита	8 бит	2^{24} (16 777 216)
double	8 байт	52 бита	11 бит	2^{53} (9 007 199 254 740 992)

Ошибки вещественных чисел.

Существует максимальный непрерывный диапазон целых чисел, представимых данным вещественным типом. Например, все целые числа до числа 16 777 216 можно представить в типе `float`, а число 16 777 217 — нельзя.
Докажите!

Константы.

- 12.3 — тип `double` со значением 12,3;
- 12.3f — тип `float` со значением 12,3;
- 12. — тип `double` со значением 12,0;
- 12.f — тип `float` со значением 12,0;

Научный стиль (scientific):

- 123e-1 — тип `double` со значением 12,3;
- 12e3 — тип `double` со значением 12 000;
- -12e3 — тип `double` со значением -12 000;
- 12e-3 — тип `double` со значением 0,012;
- -12e-3 — тип `double` со значением -0,012.

Целочисленные арифметические операции.

Для целых чисел определены 5 арифметических операторов:

- $a + b$ сложение чисел a и b ;
- $a - b$ вычитание чисел a и b ;
- $a * b$ умножение чисел a и b ;
- a / b целочисленное деление a на b ;
- $a \% b$ целочисленное деление a на b .

Тип результата.

Если складываются 2 числа одного типа, то и результат будет этого же типа. Например:

- $1+2$ даст результат 3 типа `int`,
- $1ULL + 2ULL$ даст результат $3ULL$ типа `unsigned long long`.

Тип результата.

$\text{int} \rightarrow \text{unsigned int} \rightarrow \text{long long} \rightarrow \text{unsigned long long}$.

Если складываются 2 числа разных типов, то и меньший по рангу тип преобразуется к старшему. Например

- $5 + 3\text{LL}$ преобразуется к операции $5\text{LL} + 3\text{LL}$ и даст результат 8LL ;
- $5\text{LL} - 3$ преобразуется к операции $5\text{LL} - 3\text{LL}$ и даст результат 2LL ;
- $5\text{U} * 3\text{ULL}$ преобразуется к операции $5\text{ULL} * 3\text{ULL}$ даст результат 15ULL ;
- $5 + 3\text{U}$ преобразуется к операции $5\text{U} + 3\text{U}$ даст результат 8U ;
- $5\text{ULL} - 3$ преобразуется к операции $5\text{ULL} - 3\text{ULL}$ даст результат 2ULL .

Переполнение.

Все арифметические действия производятся по модулю 2^{BIT} .
При вычитании большего из меньшего, может получиться положительное число.

- $0U - 1U$ даст результат $4294967295U$.

При сложении двух положительных знаковых чисел, может получиться отрицательное:

- $2000000000 + 1000000000$ даст результат -1294967296

(результат дает такой же остаток что и 3000000000 при делении на 2^{32}).

Оператор деления.

Целая часть и остаток.

- $17 / 5$ даст результат 3,
- $17 \% 5$ даст результат 2.

А если поделить:

- $5 / 17?$
- $5 \% 17?$
- $(-7) / 2?$
- $(-7) \% 2?$

Вещественные арифметические операции.

Для вещественных чисел определены 4 арифметических операторов:

- $a + b$ сложение чисел a и b ;
- $a - b$ вычитание чисел a и b ;
- $a * b$ умножение чисел a и b ;
- a / b целочисленное деление a на b .

Тип результата.

Любой вещественный тип имеет ранг выше целочисленного.

- $5.0 + 3LL$ преобразуется к операции $5.0 + 3.0$ и даст результат 8.0 ;
- $5.f - 3$ преобразуется к операции $5.f - 3.f$ и даст результат $2.f$;
- $5e-1 * 3ULL$ преобразуется к операции $0.5 * 3.0$ даст результат 1.5 ;
- $5 / 3.0$ преобразуется к операции $5.0 / 3.0$ даст результат около 1.66667 .

Интересные ошибки.

- `5 % 3.0?`
- `16777216.f + 1.f?`

Приоритет.

- Операторы одного приоритета выполняются слева направо.
- Приоритет операторов умножения и деления ($*$ / $%$) выше, чем приоритет сложения и вычитания $+$ $-$.
- Оператор деления $%$ имеет абсолютно такой же приоритет, что и $/$ и $*$.

Примеры.

- $1 / 3 + 2 / 3 + 3 / 3$
- $(1 + 2 + 3) / 3$
- $1.0 / 3 + 2 / 3.0 + 3.0 / 3.0$
- $12345 \% 100 / 10$
- $12345 / 10 \% 10$
- $16 / 5 - 16 \% 5 + 16 * 5$

Примеры.

преобразуется к

- $(1 / 3) + (2 / 3) + (3 / 3) = 0 + 0 + 1 = 1$
- $((1 + 2) + 3) / 3 = (3 + 3) / 3 = 6 / 3 = 2$
- $(1.0 / 3.0) + (2.0 / 3.0) + (3.0 / 3.0) = 0.33 + 0.67 + 1.00 = 2.00$
- $(12345 \% 100) / 10 = 45 / 10 = 4$
- $(12345 / 10) \% 10 = 1234 \% 10 = 4$
- $((16/5) - (16\%5)) + (16*5) = (3 - 1) + 80 = 82$.

Проверочная работа:

1. Какие из данных констант больше 30000:
а) 0xBEEF; б) 033333; в) 33333LL?
2. Определить минимальный тип чисел, если требуется:
а) сложить два целых числа 2 000 000 000 и 2 000 000 000;
б) умножить два целых числа 2 000 000 и 2 000 000;
в) поделить одно вещественно число 8,0 на 16 (точно);
3. Корректное вычисление среднего арифметического чисел 3 и 2:
а) $(3 + 2) / 2$;
б) $(3.f + 2.f) / 2$;
в) $(3 + 2) / 2LL$.
4. Вычислить
а) $1LL * 2f + 0$;
б) $1 / 2 + 2U$;
в) $1e1 / 2e2 + 3.0$.
5. Определить тип и объем в байтах результатов из №4.