

Лабораторная работа 3 (0011 = 3)

Арифметика в ЭВМ и представление данных (целочисленные операции)

Версия 2024 г. — засчитывается только в 2024 г. Актуальная версия — в <https://gitlab.com/illinc/gnu-asm>

Цель работы: изучить особенности целочисленной арифметики и арифметики с плавающей запятой.

Все задания данной лабораторной работы выполняются на чистом C/C++, без использования ассемблера. Для вывода данных до и после выполнения операции необходимо использовать функции `print16()` и `print32()`, то есть *каждое* значение должно быть представлено в формах (a)–(3).

Штраф за одно пропущенное обязательное задание — 1 балл.

Задание на лабораторную работу

Задание Л3.з1. Разработайте программу на языке C++, которая расширяет значение целочисленной переменной из 16 бит до 32 бит, рассматривая числа как:

- знаковые (signed);
- беззнаковые (unsigned).

Проверьте её работу на значениях m и n (таблица Л3.1). Каждое из двух значений — как m , так и n — должно расширяться двумя способами — как знаковым, так и беззнаковым (итого четыре операции).

Варианты целочисленных значений

Таблица Л3.1

$(N^o - 1) \% 3 + 1$	Вариант
1	$m = 231, n = -33$
2	$m = 33, n = -101$
3	$m = 201, n = -59$

Расширение числа происходит, в частности, при присваивании — если размер приёмника больше размера источника. Для явного расширения в C++ можно использовать оператор `static_cast`. Источник и приёмник должны быть либо оба знаковыми, либо оба беззнаковыми, иначе — неопределённое поведение C/C++.

К знаковой/беззнаковой интерпретации m и n можно обратиться при помощи `reinterpret_cast` адресов, как и в `print16()/print32()`.

Штраф – 2 балла, если исходное число и результат расширения выводятся не `print16()` и `print32()` соответственно, а только в одной-двух формах.

Штраф – 2 балла, если для `print16()` либо `print32()` количество выводимых цифр двоичной/шестнадцатеричной формы отлично от количества битов/тетрад.

Штраф – 1 балл, если здесь или ниже вывод хотя бы одной функции — `print16()` или `print32()` — занимает более одной строки.

Бонус +1 балл, если младшая цифра каждого представления `print32()` находится под младшей цифрой соответствующего представления `print16()`.

Задание Л3.32. Разработайте программу на языке C/C++, которая выполняет над 16-битным целочисленным значением x :

- знаковое умножение на 2;
- беззнаковое умножение на 2;
- знаковое деление на 2;
- беззнаковое деление на 2;
- расчёт остатка от беззнакового деления на 16;
- округление вниз до числа, кратного 16 (беззнаковое).

а также:

- знаковый сдвиг влево на 1 бит;
- беззнаковый сдвиг влево на 1 бит;
- знаковый сдвиг вправо на 1 бит;
- беззнаковый сдвиг вправо на 1 бит;
- рассчитывает $x \& 15$;
- рассчитывает $x \& -16$.

Сопоставьте результаты — вначале на значении m , затем n (таблица Л3.1).

Перед каждой операцией переменная имеет одно и то же начальное значение (соответственно, лучше для каждой операции создавать свою копию x); и это значение должно быть легко изменяемо.

Знаковое целочисленное умножение/деление выполняется C/C++ в том случае, если оба операнда — знаковые целые числа; беззнаковое — если оба беззнаковые. Знаковые (арифметические) сдвиги выполняются C/C++, если первый операнд (сдвигаемое число) — знаковое целое, беззнаковые (логические) сдвиги — если оно беззнаковое. Соответственно, необходимо обращаться к знаковой/беззнаковой интерпретации x при помощи `reinterpret_cast` адреса `&x`, как и в `print16()`.

Задание Л3.33. Разработайте программу на языке C/C++, которая, используя только сложение, вычитание и побитовые операции, округляет целочисленное беззнаковое значение x до кратного значению D (таблица Л3.2) двумя способами:

- а) вниз;
- б) вверх.

Варианты значений

Таблица Л3.2

$(N^0 - 1) \% 5 + 1$	Вариант
1	$D = 16$
2	$D = 32$
3	$D = 64$
4	$D = 128$
5	$D = 256$

Задание Л3.34. Разработайте программу на языке C/C++, которая выполняет для 32-битной переменной x целочисленный инкремент (то есть целочисленная интерпретация x должна увеличиться на 1).

Проверьте её работу на 32-битных целочисленных значениях m и n (таблица Л3.1), 32-битных значениях с плавающей запятой a, b, c, d из таблицы Л3.3, а также на целочисленных значениях:

- 0;
- максимальное целое 32-битное значение без знака;
- минимальное целое 32-битное значение со знаком;
- максимальное целое 32-битное значение со знаком.

Исходное значение и результат напечатайте рядом, чтобы их можно было сравнить.

Варианты значений с плавающей запятой

Таблица Л3.3

$(N^0 - 1) \% 2 + 1$	Вариант
1	$a = 0, b = 1, c = 12345678, d = 123456789$
2	$a = 0, b = 1, c = 12345689, d = 123456891$

Целочисленный инкремент/декремент выполняется C/C++ в том случае, если операнд — целое число (знаковое или беззнаковое). Соответственно, необходимо обращаться к целочисленной интерпретации x при помощи `reinterpret_cast` адреса `&x`, как и в `print32()`.

В зависимости от компилятора и его настроек при попытке инкремента максимально возможного значения типа и декремента минимального значения можно

получить предупреждение о неопределённом поведении C/C++. Занесите его в отчёт. Если дополнительных проверок не производится, и операторы инкремента ++ и декремента -- в C/C++ компилируются в команды x86 *inc/dec* — их поведение вполне определено для любого целочисленного операнда, а также одинаково для знаковых и беззнаковых целых чисел.

Задание Л3.35. Рассчитайте для заданного 32-битного значения с плавающей запятой x его модуль $|x|$, используя только битовые (целочисленные) операции и преобразование указателей.

Задание Л3.36. Разработайте программу на языке C/C++, выполняющую вычисления над числами с плавающей запятой одинарной точности (*float*). Проверьте, что программа действительно работает с операндами одинарной точности, а не приводит к типу *float* окончательный результат.

Для частичной суммы гармонического ряда $S(N) = \sum_{i=1}^N \frac{1}{i} \in \mathbb{R}$ найдите две её оценки: $S_d(N)$ — последовательно складывая члены, начиная от $i = 1$ и заканчивая $i = N$ («наивный» порядок), и $S_a(N)$ — от $i = N$ к $i = 1$. Сравните $S_d(N)$ и $S_a(N)$ для различных значений N : 10^3 , 10^6 , 10^9 . Объясните результат.

Измените тип операндов на *double*. Для вывода результата используйте *print64()* (если она не реализована — обязательно выведите, кроме *double*- интерпретации, также и *long long*-интерпретацию результата в шестнадцатеричном представлении). Объясните результат.

Л3.1. Дополнительные бонусные и штрафные баллы

—3 балла за утечку памяти (выделенные, но не освобождённые блоки динамической памяти).

Л3.2. Ссылки на теоретические сведения

2. Представление данных

Л3.3. Вопросы

1. Что такое расширение чисел со знаком и без знака? Для чего нужны операции расширения?
2. Как выполняются логические операции, побитовые операции и сдвиги над строкой битов?
3. Как представляются в памяти компьютера числа с плавающей запятой?