**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ИС**

**отчет**

**по лабораторной работе № 1**

**по дисциплине «Программирование»**

**Тема: Типы данных и их внутреннее представление в памяти**

**Студентка группы 1324 Прищепа А.С.**

**Преподаватель Глущенко А.Г.**

**Санкт-Петербург**

**2021**

1

**Цель работы:** Ознакомиться с внутренним представлением различных типов данных, используемых компьютером при их обработке; научиться работать с побитовыми операциями.

**Основные теоретические положения.**

Внутреннее представление величин целого типа – целое число в двоичном коде. При использовании спецификатора signed старший бит числа интерпретируется как знаковый (0 – положительное число, 1 – отрицательное). Для кодирования целых чисел со знаком применяется прямой, обратный и дополнительный коды.

Представление положительных и отрицательных чисел в прямом, обратном и дополнительном кодах отличается. В прямом коде в знаковый разряд помещается цифра 1, а в разряды цифровой части числа – двоичный код его абсолютной величины.

Обратный код получается инвертированием всех цифр двоичного кода абсолютной величины, включая разряд знака: нули заменяются единицами, единицы – нулями. Прямой код можно преобразовать в обратный, инвертировав все значения всех битов (кроме знакового).

Дополнительный код получается образованием обратного кода с последующим прибавлением единицы к его младшему разряду.

2

Увидеть, каким образом тип данных представляется на компьютере, можно при помощи логических операций: побитового сдвига (<<) и поразрядной конъюнкции (&).

Putchar возвращает один символ в консоль. Альтернатива - cout. В представленном способе, маска - то, с чем сравнивается значение. И побитовый сдвиг применяется для value. Таким образом 1 бит будет сравниваться с каждым битом числа. Альтернатива - побитовый сдвиг вправо, но при этом нужно проводить данную операцию не над значением(единицей), а над маской (исходым числом, битовое представление которого нужно получить).

При сдвиге вправо для чисел без знака позиции битов, освобожденные при операции сдвига, заполняются нулями. Для чисел со знаком бит знака используется для заполнения освобожденных позиций битов. Другими словами, если число 25 является положительным, используется 0, если число является отрицательным, используется 1. При сдвиге влево позиции битов, освобожденных при операции сдвига, заполняются нулями. Сдвиг влево является логическим сдвигом (биты, сдвигаемые с конца, отбрасываются, включая бит знака).

Вещественные типы данных хранятся в памяти компьютера иначе, чем целочисленные. Внутреннее представление вещественного числа состоит из двух частей – мантиссы и порядка.

3

Для 32-разрядного процессора для float под мантиссу отводится 23 бита, под экспоненту – 8, под знак – 1. Для double под мантиссу отводится 52 бита, под экспоненту – 11, под знак – 1.

Увидеть, каким образом вещественные типы данных представляются в компьютере немного сложнее. Логические операции, которые использовались с int, для вещественных типов данных не подходят. Но это ограничение можно легко обойти, использовав объединения.

Объединения – это две или более переменных расположенных по одному адресу (они разделяют одну и ту же память). Объединения определяются с использованием ключевого слова union. Объединения не могут хранить одновременно несколько различных значений, они позволяют интерпретировать несколькими различными способами содержимое одной и той же области памяти.Объедения будут служить инструментом для работы с float и double.

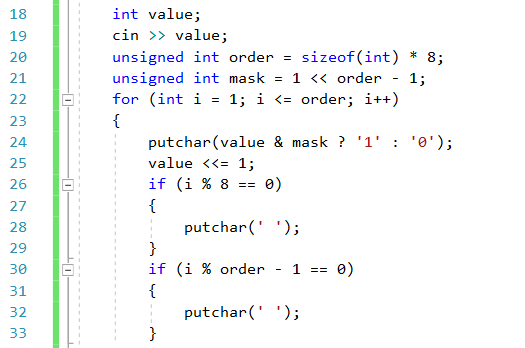
Подобные манипуляции возможны благодаря тому, что int и float занимают 4 байта. Проводя манипуляции над tool, мы изменяем значение numb\_f. Таким образом, алгоритм, который использовался для представления в памяти int может использоваться и для float.

4

Алгоритма представления double немного отличается. Под вещественное число с двойной точностью отводиться 8 байт, в то время как под int всего 4 байта. Но и это ограничение можно легко обойти. Так как данные любой линейной структуры в память записываются последовательно (друг за другом), можно использовать массив из двух int, под который будет отведено 8 байт.

**Ход работы:**

1. На экран выводится, сколько памяти (в байтах) на компьютере отводится под различные типы данных со спецификаторами и без;
2. Вводится целое число, программа считывает его;
3. Для получения двоичного представления целого числа используется логические операции (побитовый сдвиг и поразрядная конъюнкция), проводимые над числом и маской. Старший бит маски приравнивается к 1 и сдвигается вправо (циклически).В каждом повторе цикла производится поразрядная конъюнкция маски и числа и в возвращаемое значение записывается 1 или 0 (в зависимости от результата конъюнкции).

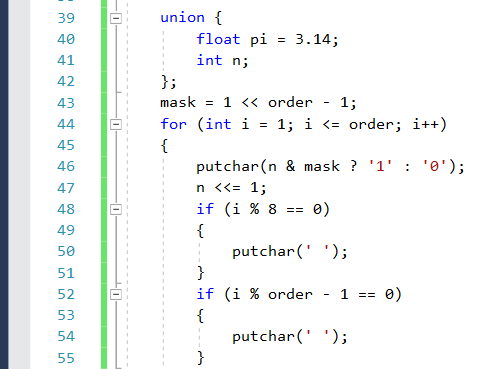


5

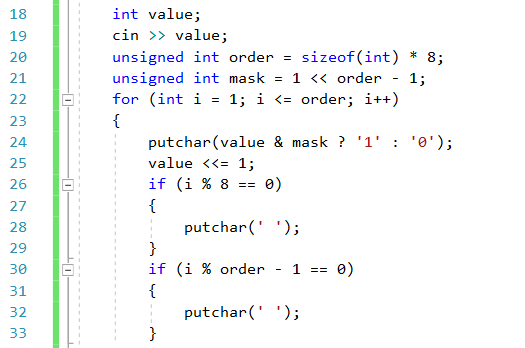
1. При получении двоичного представления с плавающей точкой используется тип данных union, содержащий в себе переменную типа int и float, которой присваивается значение числа с плавающей точкой (pi).

Двоичное представление возможно считать целочисленной переменной и получить искомое двоичное представление переменной типа float, благодаря тому, что эти типы данных имеют один размер (4 байта) и разделяют один участок памяти (из-за union).

Нахождение двоичного представления числа типа double отличается от нахождения двоичного представления числа типа float тем, что число типа double занимает в 2 раза больше места в памяти, из=за чего целочисленная переменная в union заменяется целочисленным массивом из 2 элементов, двоичные представления которых и являются возвращенным значением.

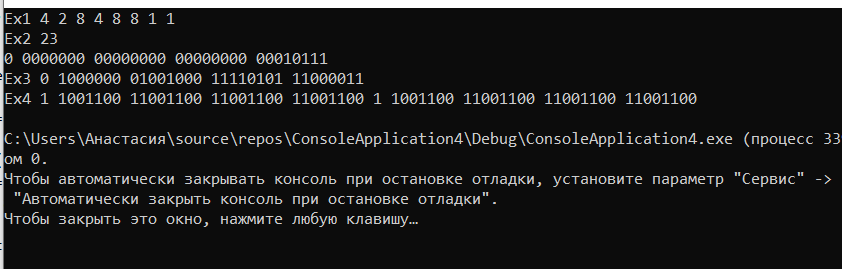


6



**Результаты работы программы:**

* при введенном числе 23



**Вывод.**

Была произведена работа с памятью переменных с помощью побитовых операций. Полученные значения схожи с теоретическими положениями, что дает возможность использовать их в дальнейшем. Созданная программа способна выводить на экран представление в памяти чисел различного типа (целого, вещественного).

7