**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №1**

**по дисциплине «Программирование»**

Тема: Типы данных и их внутреннее представление в памяти

|  |  |  |
| --- | --- | --- |
| Студент группы № 2372 |  | Юрин А. А. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2022

**Цель работы.**

Изучение числовых типов данных и их внутреннего представления в памяти компьютера. Написание программы, показывающей объём памяти, занимаемый разными типами данных и визуализирующей представление чисел в памяти.

**Основные теоретические положения.**

Представление положительных и отрицательных числе 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. Прямой код можно преобразовать в обратный, инвертировав все значения битов.

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

Увидеть, как тип данных представляется на компьютере, можно при помощи побитового сдвига и поразрядной конъюнкции.

При сдвиге вправо для положительных чисел освобожденные позиции битов, заполняются нулями. Для отрицательных – единицами. Сдвиг влево является логическим сдвигом (биты, сдвигаемые с конца, отбрасываются, включая бит знака).

Внутреннее представление вещественного числа состоит из двух частей – мантиссы и порядка (экспоненты). 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.

**Постановка задачи.**

Разработать алгоритм и написать программу, которая позволяет:

1. Вывести, сколько памяти (в байтах) отводится под различные типы данных со спецификатором и без: int, short int, long int, float, double, long double, char, bool.
2. Вывести на экран двоичное представление в памяти (все разряды) целого числа. При выводе необходимо визуально обозначить знаковый разряд и значащие разряды.
3. Вывести на экран двоичное представление в памяти (все разряды) типа float. При выводе необходимо визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если есть), мантиссу и порядок.
4. Вывести на экран двоичное представление в памяти (все разряды) типа double. При выводе необходимо визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если ест), мантиссу и порядок.

**Выполнение работы.**

Код программы представлен в приложении 1.

1. При запуске программы ожидается ввод команды с клавиатуры для запуска цикла (рис. 1).

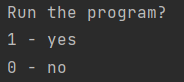


Рисунок 1. Запуск программы

1. Следующий шаг зависит от введенной команды, если пользователь ввёл:
   1. Если пользователь ввёл “0”, то выполнение программы завершается.
   2. Если “1”, то выводится строка предлагающая выбрать задание.
2. Если пользователь введёт “1”, то выводятся объёмы памяти, отводимые под различные типы данных со спецификатором (рис. 2).

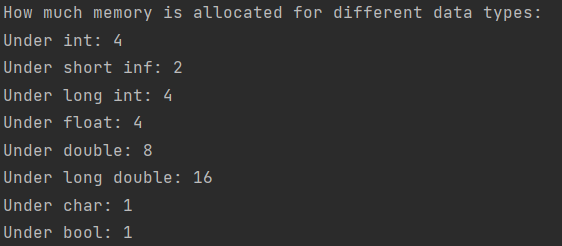


Рисунок 2. Задача I

1. Если пользователь введёт “2”, то ожидается ввод целого числа (int), двоичное представление которого потом будет выведено на экран (рис. 3).

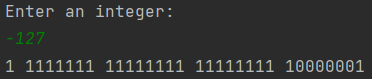


Рисунок 3. Задача II

1. Если пользователь введёт “3”, то ожидается ввод вещественного числа типа float, двоичное представление которого потом будет выведено на экран (рис. 4).

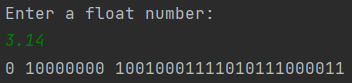


Рисунок 4. Задача III

1. Если пользователь введёт “4”, то ожидается ввод вещественного числа типа double, двоичное представление которого потом будет выведено на экран (рис. 5).

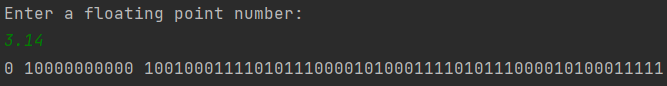


Рисунок 5. Задача IV

1. Если будет введена любая другая команда, то будет выведено на экран

(рис. 6)

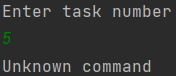


Рисунок 6.

**Выводы.**

В ходе работы было изучено представление числовых типов данных в памяти компьютера, побитовые операции сдвига и поразрядной конъюнкции, методы перевода целых и вещественных чисел в двоичный код.

Приложение 1

код

#include <iostream>  
#include <cmath>  
using namespace std;  
  
int main() {  
 setlocale(LC\_ALL, "Rus");  
 do {  
 int hello;  
 cout << "Run the program?\n" << "1 - yes\n" << "0 - no\n";  
 cin >> hello;  
 if (hello == 0) {  
 break;  
 }  
 unsigned x;  
 cout << "Enter task number\n";  
 cin >> x;  
 switch (x) {  
 case 1: {  
 cout << "How much memory is allocated for different data types: " << "\n";  
 cout << "Under int: " << sizeof(int) << "\n";  
 cout << "Under short inf: " << sizeof(short) << "\n";  
 cout << "Under long int: " << sizeof(long) << "\n";  
 cout << "Under float: " << sizeof(float) << "\n";  
 cout << "Under double: " << sizeof(double) << "\n";  
 cout << "Under long double: " << sizeof(long double) << "\n";  
 cout << "Under char: " << sizeof(char) << "\n";  
 cout << "Under bool: " << sizeof(bool) << "\n";  
 break;  
 }  
  
 case 2: {  
 int value;  
 unsigned order = sizeof(int) \* 8, mask = pow(2, 31);  
 cout << "\n" << "Enter an integer: " << "\n";  
 cin >> value;  
 for (int i = 1; i <= order; ++i) {  
 putchar(value & mask ? '1' : '0');  
 mask >>= 1;  
 if (i % 8 == 0) {  
 cout << (' ');  
 }  
 if (i % order - 1 == 0) {  
 cout << (' ');  
 }  
 }  
 cout << endl;  
 break;  
 }  
  
 case 3: {  
 union{float numb\_f;int value;};  
 unsigned order = sizeof(float) \* 8, mask = 1 << (order - 1);  
 cout << "\nEnter a float number: \n";  
 cin >> numb\_f;  
 for (int i = 1; i <= order; ++i) {  
 if (i == 2 || i == 10) {  
 cout << (' ');  
 }  
 putchar(value & mask ? '1' : '0');  
 mask >>= 1;  
 }  
 cout << endl;  
 break;  
 }  
  
 case 4: {  
 union {double numb\_d;int values[2];};  
 unsigned order = sizeof(double) \* 8, mask = 1 << (order - 1);  
 cout << "\nEnter a floating point number: \n";  
 cin >> numb\_d;  
 for (int j = 1; j >= 0; j--) {  
 for (int i = 1; i <= order / 2; i++) {  
 putchar(values[j] & mask ? '1' : '0');  
 values[j] <<= 1;  
 if ((i == 1 || i == 12) && j == 1) {  
 cout << (' ');  
 }  
 }  
 }  
 cout << endl;  
 break;  
 }  
 default: {  
 cout << "Unknown command\n";  
 }  
 }  
 } while (true);  
}