**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 гр. 2373 | Минин М.А. |  |
| Преподаватель | Глущенко А. Г. |  |

Санкт-Петербург

2022

**Цель работы.**

Изучение внутреннего представления памяти. Получение практических навыков работы с встроенными типами данными и объединениями.

**Основные теоретические положения.**

В зависимости от используемого компилятора встроенные типы данных могут занимать разное количество байт в памяти. Чтобы узнать сколько байт весит тот или иной тип данных можно использовать встроенную функцию sizeof(type\_name), которая возвращает размер типа. Для побитовых операций есть специальные операторы (>>, <<, |, &, ^), которые позволяют делать некие поразрядные операции. Ключевое слово union позволяет создавать объединения объектов, после которого они будут занимать одну область в памяти.

**Постановка задачи.**

* 1) Вывести, сколько памяти (в байтах) на вашем компьютере отводится под различные типы данных со спецификаторами и без: int, short int, long int, float, double, long double, char и bool.
* 2) Вывести на экран двоичное представление в памяти (все разряды) целого числа. При выводе необходимо визуально обозначить знаковый разряд и значащие разряды отступами или цветом.
* 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 есть), мантиссу и порядок. (\*)

**Выполнение работы.**

Код программы.

#include <iostream>

using namespace std;

int main()

{

//Задание 1

setlocale(LC\_ALL, "Russian");

cout << "Задание 1 \n\n";

cout << "Под int отводится " \

<< sizeof(int) << " байта \n";

cout << "Под short int отводится " \

<< sizeof(short int) << " байта \n";

cout << "Под long int отводится " \

<< sizeof(long int) << " байта \n";

cout << "Под float отводится " \

<< sizeof(float) << " байта \n";

cout << "Под double отводится " \

<< sizeof(double) << " байт \n";

cout << "Под long double отводится " \

<< sizeof(long double) << " байт \n";

cout << "Под char отводится " \

<< sizeof(char) << " байт \n";

cout << "Под bool отводится " \

<< sizeof(bool) << " байт \n\n";

//Задание 2

int chislo, i;

unsigned int kol\_razryadov = sizeof(chislo) \* 8;

unsigned int maska = 1 << (kol\_razryadov - 1);

cout << "Задание 2 \n\n";

cout << "Введите число int: ";

cin >> chislo;

cout << "\n" << "Двоичное представление числа " << chislo << " : \n";

for (i = kol\_razryadov - 1; i >= 0; i -= 1)

{

putchar(chislo & maska ? '1' : '0');

maska >>= 1;

if (i % 8 == 0)

putchar(' ');

if (i % (kol\_razryadov - 1) == 0)

putchar(' ');

}

//Задание 3

cout << "\n\n" << "Задание 3" << "\n\n";

cout << "Введите число float: ";

union

{

float chislo\_float;

int chislo\_3;

};

cin >> chislo\_float;

cout << "\n" << "Двоичное представление числа " << chislo\_float << " : \n";

kol\_razryadov = sizeof(chislo\_float) \* 8; // Обновляем количество разрядов

maska = 1 << kol\_razryadov - 1; // Обновляем маску

cout << "\n" << " |<--------Мантисса------->|<Порядок>|\n";

for (i = 0; i < kol\_razryadov-1; i += 1)

{

putchar(chislo\_3 & maska ? '1' : '0');

chislo\_3 <<= 1;

if (i % 8 == 0)

putchar(' ');

}

//Задание 4

cout << "\n\n" << "Задание 4" << "\n\n";

cout << "Введите число double: ";

union {

double chislo\_double;

int chislo\_4[2];

};

cin >> chislo\_double;

kol\_razryadov = sizeof(double) \* 8; //Обновляем количество разрядов для работы

maska = 1 << kol\_razryadov-1; // Обновялем маску

cout << "\n" << " |<-Порядок>|<------------------------Мантисса------------------------>|\n";

for (int i = 0; i < kol\_razryadov/2; i++)

{

putchar(chislo\_4[1] & maska ? '1' : '0');

chislo\_4[1] <<= 1;

if (i % 8 == 0 || i == 12)

putchar(' ');

} // Работает с первыми 4 битами

for (int i = 0; i < kol\_razryadov/2; i++) {

if (i % 8 == 0)

putchar(' ');

putchar(chislo\_4[0] & maska ? '1' : '0');

chislo\_4[0] <<= 1;

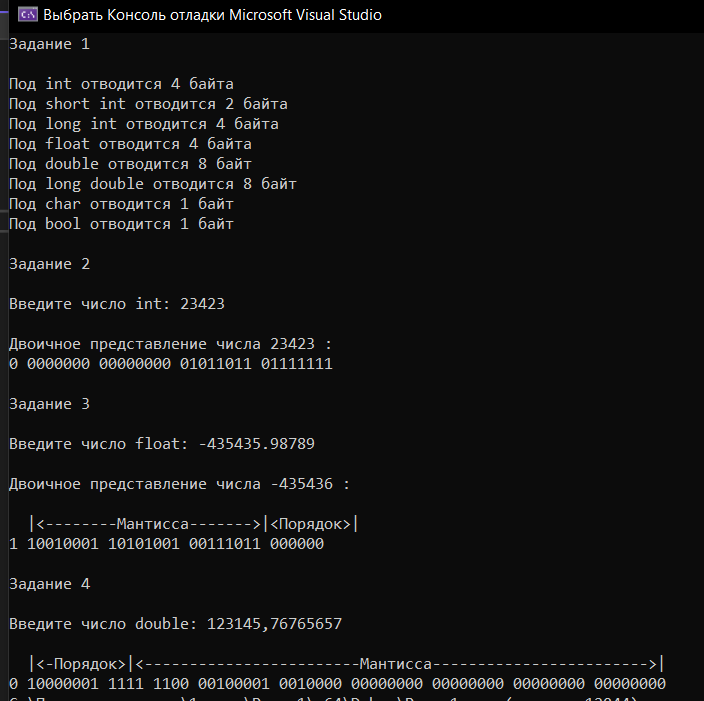
} // Работает со вторыми 4 битами

}

Блок описания кода и использованных алгоритмов

1. Используя функцию sizeof() выводим размеры требуемых типов подряд
2. Тип int весит 32 бита. Создаём другую переменную такого же размера. Использую побитовый сдвиг приводим её в вид 100…00 в двоичном представлении. И в цикле 32 раза проверяем первый разряд исходного числа с помощью побитового и(&), а потом делаем сдвиг, чтобы проверить следующий разряд.
3. У типов с плавающей точкой не перегружены операторы побитовых операций, поэтому создаём объединение этой переменной с пустой переменной целого типа, того же размера. Теперь можно выполнить алгоритм описанные в пункте 2, с целочисленной переменной из объединения и все эти операции будут отражаться на значении переменной с плавающей точкой

Блок скриншотов работы программы



**Выводы.**

Побитовые операции позволяют исследовать и работать с двоичным представлением переменных. Объединения позволяют работать со значениями переменных, с помощью операторов, которые у этих типов не перегружены.