**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

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1324 |  | Югай Н.Б. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2021

**Цель работы.**

Изучение типов данных; получение практических навыков работы с различными типами данных и представлений о том, как они представлены в памяти.

**Основные теоретические положения.**

Внутреннее представление величин целого типа – целое число в

двоичном коде. Для кодирования целых чисел со знаком применяется несколько видов кодов:

1. Прямой — в знаковый разряд помещается цифра 1 или 0, в зависимости от знака данного числа, а в разряды цифровой части числа – двоичный код его абсолютной величины;
2. Обратный — получается путём инвертирования всех цифр двоичного кода, получаемых прямым способом;
3. Дополнительный коды — получается прибавлением единицы к младшему разряда обратного кода.

Внутреннее представление вещественного числа состоит из двух

частей – мантиссы и порядка. Для 32-разрядного процессора для float под

знак уходит 1 бит, под экспоненту – 8, а мантиссу – 23. Для double же отводится 1, 11 и 52 бита соответственно.

**Постановка задачи.**

Необходимо создать код, который будет:

1) выводить количество памяти (в байтах), выделяемую на следующие типы данных: int, short int, long int, float, double, long double, char и bool;

2) выводить на экран двоичное представление в памяти вводимого целого числа;

3) выводить на экран двоичное представление в памяти вводимого вещественного числа типа float.

**Выполнение работы.**

Код программы представлен в приложении А.

В лабораторной работе были применены оператор sizeof, для вычисления отводимого объёма памяти под каждый из типов данных, и тернарный оператор, для создания двоичного представления.

Для перевода целого числа в бинарный вид создаётся маска вида 100..0, с количеством цифр, равных отводимому количеству битов под определённый тип данных. Для определения того, стоит на текущем месте 1 или 0, используется оператор побитового И. После завершения сравнения, программа выводит 1 или 0, в зависимости от полученного результата, и производит побитовый сдвиг влево в исходном числе. Цикл продолжается, пока не обработает последний бит исходного числа.

Для перевода вещественного числа в бинарный вид используется похожий алгоритм, с той лишь разницей, что для обхода отсутствия возможности производить побитовые операции с вещественными типами данных используется группировка переменных union, которая даёт возможность обращаться к одной и той же переменной и как к целому, и как к вещественному числу. В этом случае число вводится в “вещественное представление”, а сравнивается в “целом”, также производя побитовый сдвиг влево до конца.

Также для разделения бита, отводимого под знак, мантиссы и экспонента в коде прописано написание пробела после определённых битов.

|  |  |
| --- | --- |
| При запуске программы перед пользователем появляется окно с количеством байт, которые отводятся на каждый из перечисленных типов данных, и предложение ввести целое число для обработки его в двоичное представление |  |

|  |  |
| --- | --- |
| При вводе целого числа программа выводит следующей строкой его побитовое представление, а также предлагает ввести уже вещественное число |  |
| При вводе целого числа программа выводит следующей строкой его побитовое представление, а также делает паузу |  |

|  |  |
| --- | --- |
| При нажатие любой из клавиш программа заканчивает работу |  |

**Выводы.**

В ходе лабораторной работы были изучение типов данных и написана программа для представления их в бинарном коде.

Приложение А

рабочий код

﻿#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

cout << "int -- " << sizeof(int) << endl;

cout << "short int -- " << sizeof(short int) << endl;

cout << "long int -- " << sizeof(long int) << endl;

cout << "float -- " << sizeof(float) << endl;

cout << "double -- " << sizeof(double) << endl;

cout << "long double -- " << sizeof(long double) << endl;

cout << "char -- " << sizeof(char) << endl;

cout << "bool -- " << sizeof(bool) << endl;

//--------------------------------------------//

unsigned int bytes = 4;

unsigned int amount = bytes \* 8;

unsigned int mask = 1 << (amount - 1);

int value;

//--------------------------------------------//

cout << "Введите целое число: ";

cin >> value;

cout << "Побитовое представление: ";

for (int i = 1; i <= amount; i++)

{

cout << ((value & mask) ? '1' : '0');

value <<= 1;

if (i == 1) { cout << ' '; }

}

cout << endl;

//--------------------------------------------//

union

{

int value;

float number;

} Float;

cout << "Введите число float: ";

cin >> Float.number;

cout << "Побитовое представление: ";

for (int i = 1; i <= amount; i++)

{

cout << ((Float.value & mask) ? '1' : '0');

Float.value <<= 1;

if (i == 9 || i == 1) { cout << ' '; }

}

system("pause");

return 0;

}