**ОТЧЕТ**

**по лабораторной работе №1**

**по дисциплине****«Организация ЭВМ и систем»**

**Тема: исследование внутреннего представления различных форматов данных**

Выполнил: Гаврилов П.А.

Групппа: 6891

**Цель работы**

Знакомство с внутренним представлением различных типов данных, используемых компьютером при их обработке.

**Задачи**

1. Разработать алгоритм ввода с клавиатуры данных с типом unsigned int и double и показать на экране их внутреннее представление в двоичной системе счисления;

2. Написать и отладить программу на языке С++, реализующую разработанный алгоритм;

3. Дополнить разработанный ранее алгоритм блоками для выполнения преобразования (установить в заданное пользователем состояние определённое количество рядом стоящих бит, номер младшего из которых, как и всё остальное, вводится с клавиатуры) двоичного полученного кода исходного типа данных и последующего вывода преобразованного кода в двоичной системе счисления и в формате исходного данного.

**Текст программы**

**#include <iostream>**

#include **<bitset>**

**template**<**typename** T>

**int** run();

**void** showNoArgError();

**void** showWrongArgError(**const char** \*arg);

**void** showError();

**template**<**typename** T>

T getNumber();

**using namespace** std;

**template**<**typename** T>

**union** Utype {

Utype(T t) : t(t) {}

Utype(**unsigned long** u) : u(u) {}

T t;

**unsigned long** u;

};

**int** main(**int** argc, **char** \*argv[]) {

**if** (argc == 1) {

showNoArgError();

**return** -1;

} **else** {

**const char** \*arg = argv[1];

**if** (string(arg) == **"uint"**) {

**return** run<u\_int>();

} **else if** (string(arg) == **"double"**) {

**return** run<**double**>();

} **else** {

showWrongArgError(arg);

**return** -1;

}

}

}

**void** showWrongArgError(**const char** \*arg) {

cout << **"Указан неверный аргумент: "** << arg << endl;

showNoArgError();

}

**void** showNoArgError() {

cout << **"Для запуска укажите аргумент: <uint> - для unsigned int, <double> - для double"** << endl;

}

**template**<**typename** T>

**int** run() {

**const int** typeLength = **sizeof**(T) \* 8;

**const int** typeMaxBitNumber = typeLength - 1;

T number(0);

cout << **"Введите число: "**;

number = getNumber<T>();

cout << **"Вы ввели: "** << number << endl;

Utype<T> utype(number);

bitset<typeLength> bits(utype.u);

cout << **"Введенное число в двоичном виде: "** << endl;

cout << bits << endl;

u\_short position(0);

cout << **"Введите номер младшего бита от ["**

<< 0

<< **"] до ["**

<< typeMaxBitNumber

<< **"]"** << endl;

position = getNumber<u\_short>();

**if** (position < 0 || position > typeMaxBitNumber) {

showError();

**return** -1;

}

u\_short count(0);

cout << **"Введите количество битов для изменения от ["**

<< 0

<< **"] до ["**

<< typeMaxBitNumber - position + 1

<< **"]"** << endl;

count = getNumber<u\_short>();

**if** (count < 0 || count > typeMaxBitNumber - position + 1) {

showError();

**return** -1;

}

**int** bit;

**for** (size\_t i = position; i < position + count; ++i) {

cout << i

<< **"-й бит: "**;

bit = getNumber<**int**>();

**if** (bit != 0 && bit != 1) {

showError();

**return** -1;

}

bits.set(i, **static\_cast**<**bool**>(bit));

}

cout << **"Полученное число в двоичном виде: "** << endl;

cout << bits << endl;

cout << **"Полученное число в десятичном виде: "** << endl;

Utype<T> result(bits.to\_ulong());

cout << result.t << endl;

**return** 0;

}

**template**<**typename** T>

T getNumber() {

T result;

cin >> result;

**if** (cin.fail()) {

showError();

exit(-1);

}

**return** result;

}

**void** showError() { cout << **"Ошибка!!!"** << endl; }

**Результаты работы программы**

Ниже представлены скриншоты консольного вывода результатов работы программы с двумя разными типами данных.

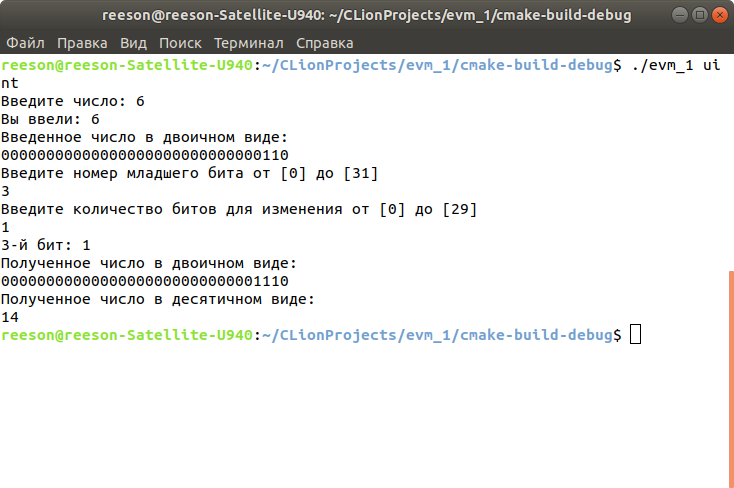


Рисунок 1 – Пример работы программы u\_int



Рисунок 4 – Пример работы программы (2)

**Схема аппаратных средств**

На рисунке 5 представлена структурная схема аппаратных средств, используемых при выполнении программы. Системная шина передает информацию между процессором и остальными устройствами ЭВМ. Системная шина состоит из трех шин: шины управления, шины данных и адресной шины. По этим шинам циркулируют управляющие сигналы, данные (числа, символы), адреса ячеек памяти и номера устройств ввода-вывода.

Процессор состоит из четырех устройств: арифметико-логического устройства (АЛУ), устройства управления (УУ), блока регистров (БР) и кэш-памяти. АЛУ выполняет арифметические и логические операции над данными. Промежуточные результаты сохраняются в БР. Кэш-память служит для повышения быстродействия процессора путем уменьшения времени его непроизводительного простоя. УУ отвечает за формирование адресов очередных команд, т. е. за порядок выполнения команд, из которых состоит программа.

Некоторые результаты вычислений поступают на монитор для отображения пользователю.

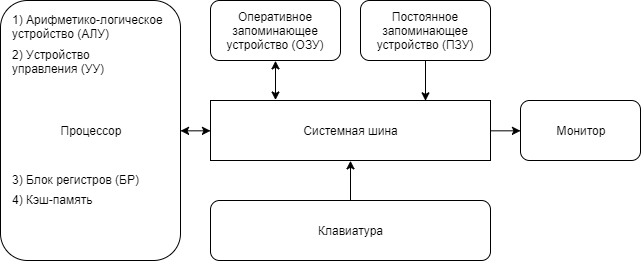


Рисунок 5 – Схема аппаратных средств

**Выводы**

В результате выполнения лабораторной была достигнута цель и выполнены задачи. Были получены практические навыки по работе с различными типами данных и их внутренним представлением.