**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №1**

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

Тема: ИССЛЕДОВАНИЕ ВНУТРЕННЕГО ПРЕДСТАВЛЕНИЯ РАЗЛИЧНЫХ ФОРМАТОВ ДАННЫХ

|  |  |  |
| --- | --- | --- |
| Студент гр. 5307 |  | Самоуков Н.В. |
| Преподаватель |  | Иванов П.В. |

Санкт-Петербург

2016

Цель работы: знакомство с внутренним представлением различных типов данных, используемых компьютером при их обработке.

Задание:

1. Разработать алгоритм ввода с клавиатуры требуемых типов данных 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 двоичного полученного кода исходного типа данных и последующего вывода преобразованного кода в двоичной системе счисления и в формате исходного данного.

Вариант №14. Данные short int и double. Обработка: инвертировать значения рядом стоящих бит, количество которых и номер старшего разряда задаются с клавиатуры.

**Схема программы:**

Схема программы представленна на рис. 1.



Рис. 1. Схема программы.

**Текст программы:**

#include <conio.h>

#include <stdio.h>

#include <windows.h>

#include <locale.h>

#include <string>

#include <vector>

using namespace std;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

class TYPE\_DATA

{

public:

string name,print\_st;

int bytes;

int p\_bytes;

byte \*buf;

TYPE\_DATA()

{

buf = nullptr;

}

~TYPE\_DATA()

{

delete[]buf;

buf = nullptr;

}

void print\_bytes()

{

string res;

for (int i = 0; i < bytes; i++)

for (int r = 0; r < 8; r++)

{

res = (char)(buf[i]&(1<<r)) + res;

}

for (int i = 0; i < res.length(); i++)

{

SetConsoleTextAttribute(hConsole, (WORD)(5));

if (i <= p\_bytes)

SetConsoleTextAttribute(hConsole, (WORD)(4));

if (i == 0)

SetConsoleTextAttribute(hConsole, (WORD)(7));

printf("%c", res[i]?'1':'0');

}

SetConsoleTextAttribute(hConsole, (WORD)(15));

}

void print\_base()

{

if (print\_st.length()>0)

{

if (bytes == 1)

printf(print\_st.data(), \*(byte\*)(buf));

if (bytes == 2)

printf(print\_st.data(), \*(short\*)(buf));

if (bytes == 4)

printf(print\_st.data(), \*(int\*)(buf));

if (bytes == 8)

printf(print\_st.data(), \*(long long\*)(buf));

}

}

void invert\_bits(int en, int count)

{

if (en >= bytes \* 8)

en = - 1;

for (int i = en; i > en - count && i >= 0; i--)

{

buf[i / 8] ^= (1 << (i % 8));

}

}

};

vector < string > split\_command(string str)

{

vector < string > res;

for (int i = 0; i < str.length(); i++)

{

if (i == 0 && str[i] == ' ')

{

str.erase(i, 1);

i--;

}

if (i>0 && i == str.length()-1 && str[i] == ' ')

{

str.erase(i, 1);

i-=2;

}

if (i > 0 && str[i] == ' ' && str[i - 1] == ' ')

{

str.erase(i, 1);

i--;

}

}

str += ' ';

string resres;

for (int i = 0; i < str.length(); i++)

{

if (str[i] == ' ')

{

res.push\_back(resres);

resres = "";

}

else

{

resres += str[i];

}

}

return res;

}

void main()

{

TYPE\_DATA number;

number.bytes = 0;

number.name = "none";

number.print\_st = "";

number.buf = nullptr;

setlocale(0, "RU");

HANDLE hCon = CreateConsoleScreenBuffer(GENERIC\_READ | GENERIC\_WRITE, 0, NULL, CONSOLE\_TEXTMODE\_BUFFER, NULL);

if (hCon != INVALID\_HANDLE\_VALUE) {

CONSOLE\_FONT\_INFOEX cfi;

cfi.cbSize = sizeof(CONSOLE\_FONT\_INFOEX);

cfi.nFont = 0;

cfi.dwFontSize.X = 0;

cfi.dwFontSize.Y = 24;

cfi.FontFamily = FF\_DONTCARE;

cfi.FontWeight = 400;

wcscpy(cfi.FaceName, L"Lucida Console");

SetCurrentConsoleFontEx(hCon, FALSE, &cfi);

}

SetConsoleTextAttribute(hConsole, (WORD)(15));

int i; //Введённое число

int i\_res; //Результат обработки числа

bool is\_input; //Флажок ввода

bool is\_res; //Флажок результата

int menu; //Переменная выбранного пункта меню

is\_input = 0;

is\_res = 0;

i = 0;

i\_res = 0;

char buf[1024];

bool is\_exit = 0;

do

{

system("cls");

printf("Число:");

number.print\_base();

printf("\n");

printf("Битовое представление:\n");

number.print\_bytes();

printf("\n");

printf("Тип:");

printf(number.name.data());

printf("\n");

puts("Комманды:");

puts("chnt-сменить тип переменной");

puts("set <n>-Сменить значение на <n>");

puts("setb <bits>-Сменить биты на <bits>");

puts("inv <en> <co>-инверсия <co> битов заканчивая битом <en>");

puts("exit-выход");

gets\_s(buf, 1024);

auto arr\_com = split\_command(buf);

if (arr\_com.size() > 0)

{

if (arr\_com[0] == "exit")

is\_exit = 1;

if (arr\_com[0] == "chnt")

{

if (number.name == "double")

{

number.name = "short int";

number.bytes = sizeof(short int);

number.buf = new byte[sizeof(short int)];

for (int i = 0; i < sizeof(short int);i++)

number.buf[i] = -1;

number.p\_bytes = 0;

number.print\_st = "%hd";

}

else

{

number.name = "double";

number.bytes = sizeof(double);

number.buf = new byte[sizeof(double)];

for (int i = 0; i < sizeof(double); i++)

number.buf[i] = 0;

number.p\_bytes = 11;

number.print\_st = "%lf";

}

}

}

if (arr\_com.size() > 1)

{

if (arr\_com[0] == "set")

{

sscanf(arr\_com[1].data(), number.print\_st.data(), number.buf);

}

if (arr\_com[0] == "setb")

{

for (int i = number.bytes \* 8 - 1; i >= 0; i--)

number.buf[i / 8] = 0;

for (int i = number.bytes \* 8 - 1, r = 0; i >= 0 && r < arr\_com[1].size(); i--, r++)

{

if (arr\_com[1][r] == '1')

number.buf[i / 8] ^= (1 << (i % 8));

}

}

}

if (arr\_com.size() > 2)

{

int pos = atoi(arr\_com[1].data());

int num = atoi(arr\_com[2].data());

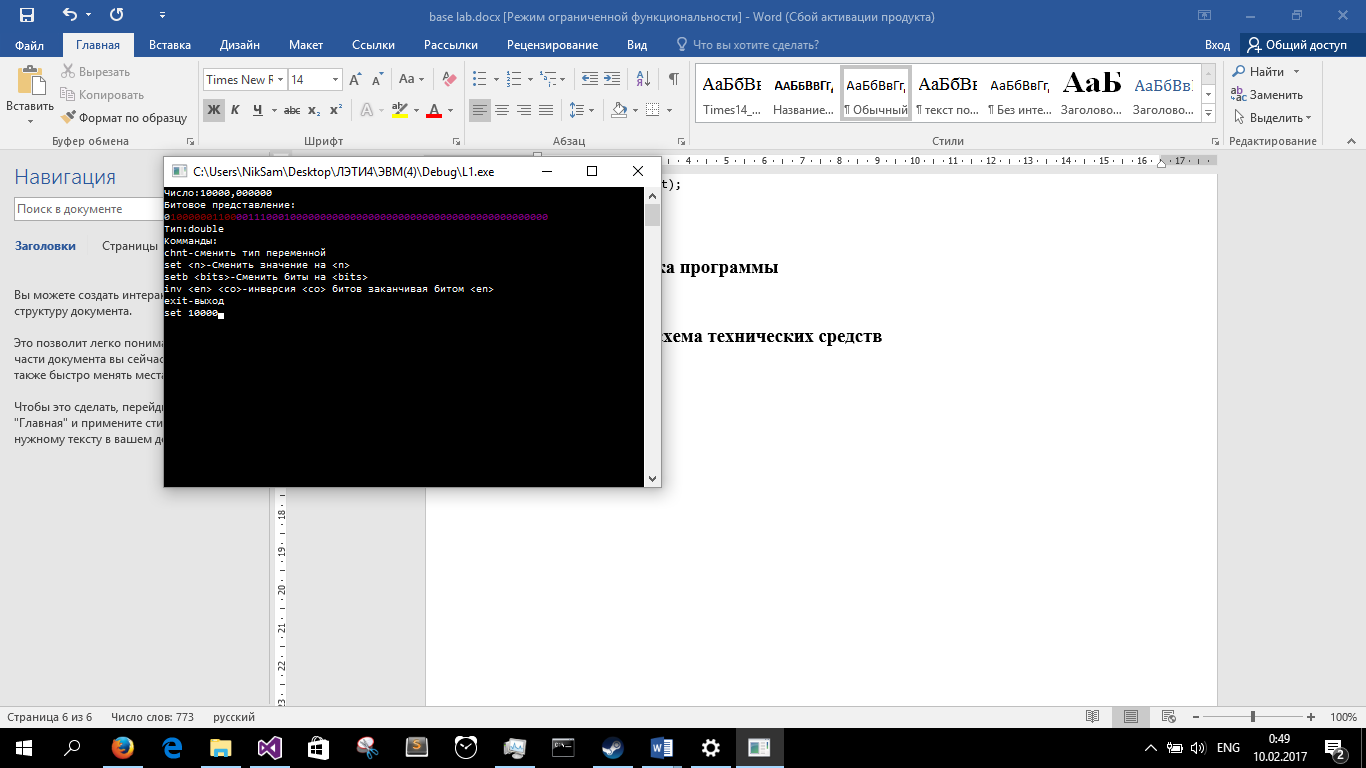
number.invert\_bits(pos, num);

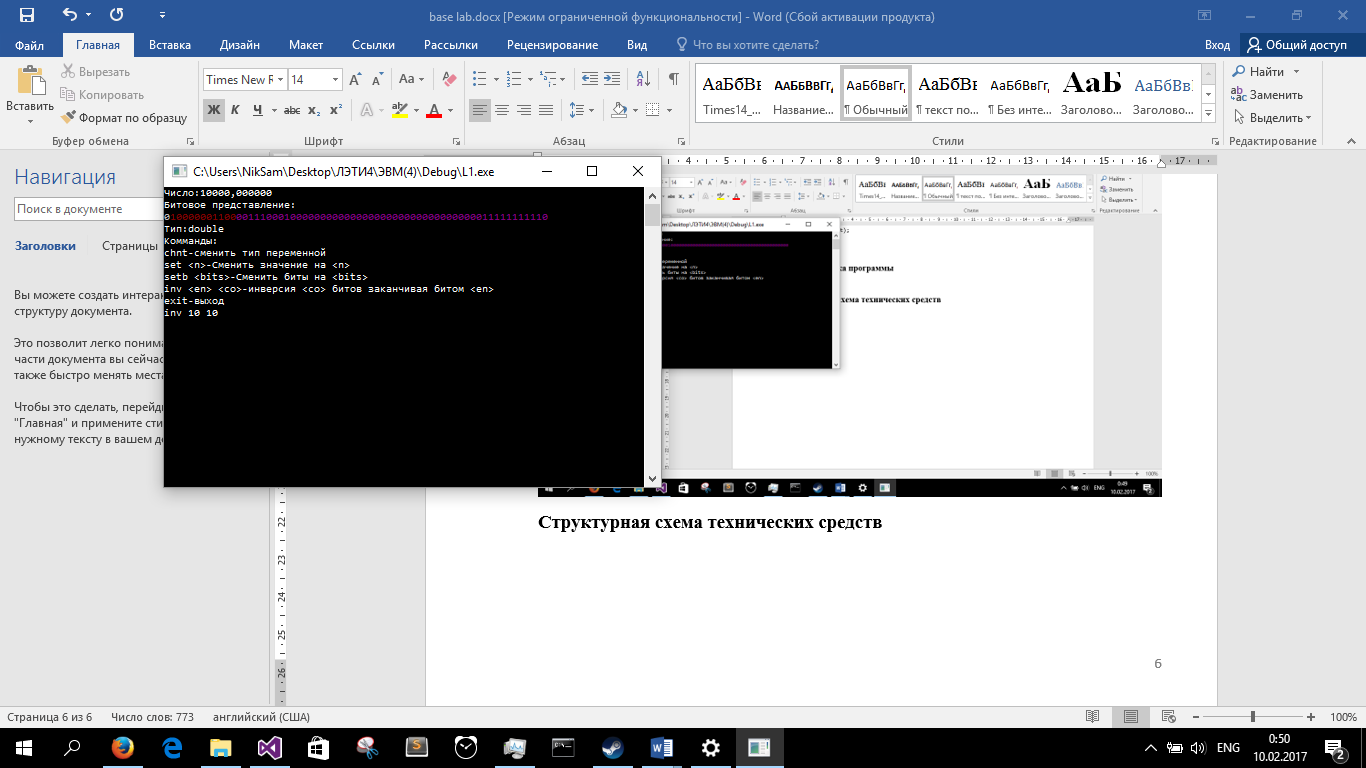
}

} while (!is\_exit);

}

**Примеры запуска программы**





**Структурная схема технических средств**

