МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Организация ЭВМ и систем»

Тема: Исследование внутреннего представления

различных форматов данных

Студенты гр. 5371 Бергер Э. Э.

Локкина О. С.

Уруков С. Д.

Преподаватель Рукавицын А. Н.

Санкт-Петербург

2017

**ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ**

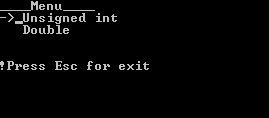
*Вариант 4*

Разработать алгоритм ввода с клавиатуры типов данных unsigned int и double и показать на экране их внутреннее представление в двоичной системе счисления. Дополнить разработанный ранее алгоритм блоками для установки в заданное пользователем состояние определённого количества рядом стоящих бит, номер младшего из которых, как и всё остальное, вводится с клавиатуры, и последующего вывода преобразованного кода в двоичной системе счисления и в формате исходного данного.

**БЛОК-СХЕМА АЛГОРИТМА**

**ПРИМЕРЫ ЗАПУСКА ПРОГРАММЫ**

При запуске программы открывается меню, позволяющее выбрать тип данных для работы:

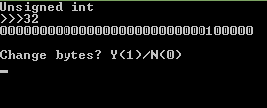


Перемещение по пунктам меню осуществляется с помощью стрелок ↑ и ↓, выход происходит при нажатии клавиши Esc, выбор пункта – при нажатии Enter.

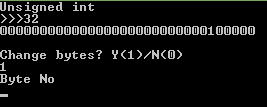
Рассмотрим несколько примеров работы программы:

1. *Unsigned int*

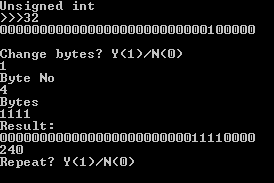
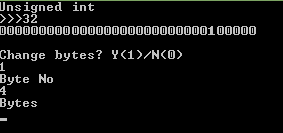
При вводе числа 32 программа выводит на экран внутреннее представление этого числа в двоичном виде.



Также дается возможность изменить биты в представлении числа. Сначала нужно ввести бит, с которого начнется изменение (биты нумеруются справа налево начиная с 0), а потом – ввести желаемые значения битов. Новые биты вставляются тоже справа налево.



Изменим биты, начиная с 4, на 1111.

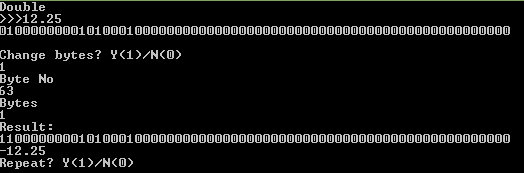


Программа выводит результат в двоичном и десятичном видах. Видно, что после изменения число 32 превратилось в 240.

Далее можно продолжить работу с типом unsigned int и ввести новое число, либо вернуться в меню.

1. *Double*

Последовательность действий при работе с числами типа double аналогична типу unsigned int. Посмотрим на результат ввода числа 12.25 и изменения его знака.



**ТЕКСТ ПРОГРАММЫ**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdafx.h>

#include <iostream>

#include <math.h>

#include <string>

#include <windows.h>

#include <conio.h>

COORD m = { 0, 0 };

COORD c = { 0, 1 };

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

union doubleStructure{

double number;

unsigned long long binary;

};

static doubleStructure doubleResult;

template<class T> std::string binaryNotation(T\* object, int countOfBytes);

template<class T> void fromBinaryToInt( T\* object, std::string bytes);

std::string replaceBytes(std::string result);

void menu();

int Point();

int main()

{

menu();

return 0;

}

template<class T> std::string binaryNotation(T\* object, int countOfBytes) {

std::string result;

bool mask = 1;

unsigned long long obj = (unsigned long long)\*object; //unsigned long long = unsigned \_\_int64 - 8 байт

for (int i = 0; i < countOfBytes; i++) {

result += std::to\_string(mask & obj);

obj >>= 1;

}

std::reverse(result.begin(), result.end());

return result;

}

template<class T> void fromBinaryToInt(T\* object, std::string bytes) { //Перевод из строки в число

for (int i = 0; i < (sizeof(T) \* 8); i++) {

\*object <<= 1;

if (bytes[i] == '1') {

\*object += 1;

}

}

}

void menu() {

int menuPoint;

system("cls");

SetConsoleCursorPosition(h, m);

std::cout << "\_\_\_\_Menu\_\_\_\_"

<< "\n Unsigned int"

<< "\n Double"

<< "\n\n\n!Press Esc for exit";

menuPoint = Point();

while (1) {

system("cls");

bool tmp;

std::string result;

switch (menuPoint)

{

case 1:

unsigned int uint;

std::cout << "Unsigned int\n>>>";

std::cin >> uint;

result = binaryNotation(&uint, sizeof(unsigned int) \* 8);

std::cout << result;

//std::cin.clear();

std::cout << "\n\nChange bytes? Y(1)/N(0)\n";

std::cin >> tmp;

if (tmp) {

result = replaceBytes(result);

uint = std::stoul(result, 0, 2); //В большинстве компиляторов unsigned long и unsigned int эквивалентны по байтам

std::cout << uint << std::endl;

}

std::cout << "Repeat? Y(1)/N(0)\n";

std::cin >> tmp;

if (!tmp) {

menu();

}

break;

case 2:

std::cout << "Double\n>>>";

std::cin >> doubleResult.number;

result = binaryNotation(&doubleResult.binary, sizeof(unsigned long long) \* 8);

std::cout << result;

//std::cin.clear();

std::cout << "\n\nChange bytes? Y(1)/N(0)\n";

std::cin >> tmp;

if (tmp) {

result = replaceBytes(result);

doubleResult.binary = std::stoull(result, 0, 2);

//fromBinaryToInt(&doubleResult.binary, result);

std::cout << doubleResult.number << std::endl;

}

std::cout << "Repeat? Y(1)/N(0)\n";

std::cin >> tmp;

if (!tmp) {

menu();

}

break;

}

}

}

std::string replaceBytes(std::string result) {

std::string replacementBytes;

short byteNumer;

std::cout << "Byte No\n";

std::cin >> byteNumer;

std::cout << "Bytes \n";

std::cin >> replacementBytes;

std::reverse(result.begin(), result.end()); //В обратном порядке

result.replace(byteNumer, replacementBytes.size(), replacementBytes);

std::reverse(result.begin(), result.end());

std::cout << "Result:\n" << result << std::endl;

return result;

}

int Point()

{

char point[] = "->";

short stop = 2;

c.Y = 1;

int menu = 1;

COORD temp = { 0, 0 };

while (1)

{

SetConsoleCursorPosition(h, c);

std::cout << " ";

SetConsoleCursorPosition(h, c);

std::cout << point;

temp.Y = c.Y;

int key = \_getch();

if (key == 0 || key == 224)

key = \_getch();

switch (key)

{

case 72:

if (c.Y != 1)

{

c.Y--;

menu--;

}

break;

case 75:

if (c.Y != 1)

{

c.Y--;

menu--;

}

break;

case 77:

if (c.Y != stop)

{

c.Y++;

menu++;

}

break;

case 80:

if (c.Y != stop)

{

c.Y++;

menu++;

}

break;

case 27:

exit(1);

case 13:

return menu;

default: break;

}

SetConsoleCursorPosition(h, temp);

std::cout << " ";

SetConsoleCursorPosition(h, c);

std::cout << point;

}

};