**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра вычислительной техники**

отчет

**по лабораторной работе №1**

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

Тема: Исследование внутреннего представления различных форматов данных

|  |  |  |
| --- | --- | --- |
| Студент гр. 2372 |  | Васильев Ю.А. |
| Преподаватель |  | Гречухин М. Н. |

Санкт-Петербург

2023

**Цель работы**

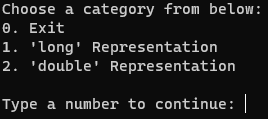
Изучение числовых типов данных и их внутреннего представления в памяти компьютера. Написание программы, показывающей объём памяти, занимаемый long и double и визуализирующей представление чисел в памяти.

**Формулировка задания**

Разработать алгоритм ввода с клавиатуры требуемых типов данных и показать на экране их внутреннее представление в двоичной системе счисления. Выполнить циклический сдвиг в заданную пользователем сторону на заданное количество разрядов в пределах определённой группы разрядов, количество которых и номер старшего разряда в группе задаются с клавиатуры.

**Результаты работы программы**

1. При старте программы предусмотрен ввод команды с клавиатуры для выполнения определенного действия.



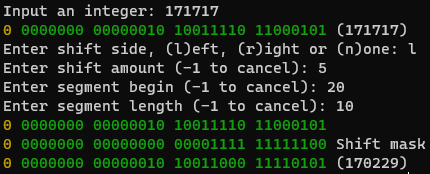
1. Следующий шаг зависит от введенной команды, если пользователь ввёл:

Если пользователь ввёл “0”, то выполнение программы завершается.

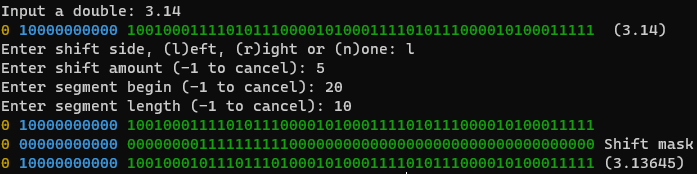
Если “1”, то пользователь входит в режим работы с типом данных ‘long’ (см п. 3)

Если “2”, то пользователь входит в режим работы с типом данных ‘double’. (см п.4)

1. Если пользователь введёт “1”, то ожидается ввод целого числа (int), также сторона сдвига (l)eft, (r)ight или (n)one – выход из данного подменю. В любое время пользователь может ввести -1 для выхода из данного подменю.



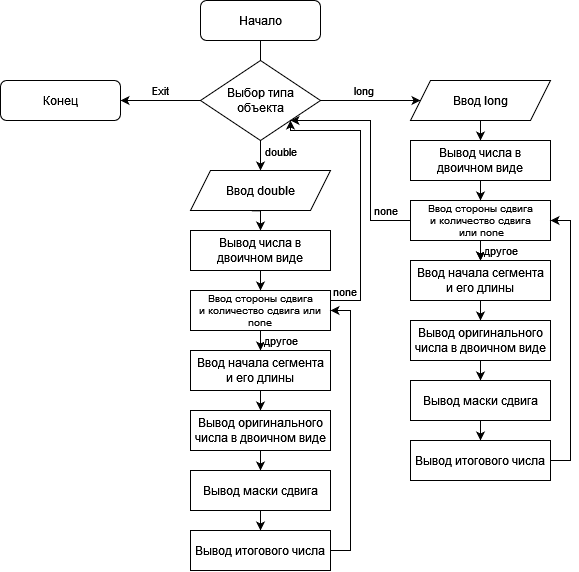
1. Если пользователь введёт “2”, то ожидается ввод целого числа (int), также сторона сдвига (l)eft, (r)ight или (n)one – выход из данного подменю. В любое время пользователь может ввести -1 для выхода из данного подменю.



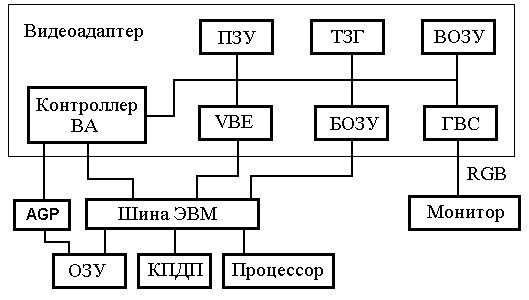
1. Если будет введена любая другая команда, то будет выведена ошибка на экран.



**Блок-схема**

****

**Структурная схема аппаратных средств**

****

**Текст программы**

#include <iostream>

#include <limits>

using namespace std;

void printLong(long);

void printDouble(long long);

void varyLong(long&, char, int, int, int);

void varyDouble(unsigned long long&, char, int, int, int);

void longTask();

void doubleTask();

/\*

\* Функция для ввода данных в терминал

\* При вызове функции нужно указать получаемые данные в скобках, т.е. readValue<int>() - получить число

\*/

template <typename T>

T readValue() {

T value{};

while (true) {

cin >> value;

if (cin.fail()) {

cout << "\nIncorrect input. Enter new value: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

else {

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

return value;

}

}

}

int main()

{

while (true) {

system("cls");

cout <<

"Choose a category from below:\n"

"0. Exit\n"

"1. 'long' Representation\n"

"2. 'double' Representation\n\n";

cout << "Type a number to continue: ";

int choice = readValue<int>();

system("cls");

switch (choice) {

case 0:

return 0;

case 1:

longTask();

break;

case 2:

doubleTask();

break;

default:

cout << "\nCategory with number " << choice << " does not exist." << endl;

}

system("pause");

}

}

/\*

Вывод long на экран в двоичном формате

\*/

void printLong(long number) {

unsigned int order = sizeof(number) \* 8;

unsigned long mask = 1ul << (order - 1);

cout << "\x1B[33m"; // Bit Sign

for (unsigned int bit = 1; bit <= order; bit++) {

cout << (bool)(number & mask);

mask >>= 1;

if (bit == 1)

cout << "\x1B[32m "; // Other Bits

else if (bit % 8 == 0)

cout << ' ';

}

cout << "\x1B[0m"; // Reset colors

}

/\*

Изменение битов long в соответствии с доп. заданием

number: long

side: r/l - сторона

amount: int - количество сдвига

begin: int - номер старшего бита

length: int - длина диапазона

\*/

void varyLong(unsigned long& number, char side, int amount, int begin, int length) {

amount = amount % length;

unsigned long mask = (~0ul >> begin) ^ (~0ul >> (begin + length)); // Mask to make row of ones in specified sector

if (begin + length == sizeof(long) \* 8)

mask = ~0;

unsigned long n = number & mask;

printLong(mask);

cout << "Shift mask" << endl;

if (side == 'r') // Cycling Shift

n = (n >> amount) | (n << (length - amount));

else

n = (n << amount) | (n >> (length - amount));

number = (number & ~mask) | (n & mask); // Applying changes

}

void longTask() {

union {

long number = 0;

unsigned long repr;

};

cout << "Input an integer: ";

number = readValue<long>();

printLong(number);

cout << "(" << number << ")" << endl;

while (true) {

char side;

int amount;

int begin;

int length;

do {

cout << "Enter shift side, (l)eft, (r)ight or (n)one: ";

side = readValue<char>();

} while (side != 'l' && side != 'r' && side != 'n');

if (side == 'n') return;

do {

cout << "Enter shift amount (-1 to cancel): ";

amount = readValue<int>();

} while (amount < -1);

if (amount == -1) return;

do {

cout << "Enter segment begin (-1 to cancel): ";

begin = readValue<int>();

} while (begin < -1);

if (begin == -1) return;

do {

cout << "Enter segment length (-1 to cancel): ";

length = readValue<int>();

} while (length < -1);

if (length == -1) return;

if (begin >= sizeof(long long) \* 8) return;

if (begin + length > sizeof(long long) \* 8) return;

printLong(number);

cout << endl;

varyLong(repr, side, amount, begin, length);

printLong(number);

cout << "(" << number << ")" << endl;

}

}

void printDouble(long long number) {

unsigned int order = sizeof(number) \* 8;

unsigned long long mask = 1ull << (order - 1);

cout << "\x1B[33m"; // Bit Sign

for (unsigned int bit = 1; bit <= order; bit++) {

cout << (bool)(number & mask);

mask >>= 1;

if (bit == 1)

cout << "\x1B[36m "; // Mantissa

if (bit == 12)

cout << "\x1B[32m "; // Exponent

}

cout << "\x1B[0m "; // Reset colors

}

/\*

Изменение битов double в соответствии с доп. заданием

number: union-фицированный double

side: r/l - сторона

amount: int - количество сдвига

begin: int - номер старшего бита

length: int - длина диапазона

\*/

void varyDouble(unsigned long long& number, char side, int amount, int begin, int length) {

amount = amount % length;

unsigned long long mask = (~0ull >> begin) ^ (~0ull >> (begin + length)); // Mask to make row of ones in specified sector

if (begin + length == sizeof(long long) \* 8)

mask = ~0;

unsigned long long n = number & mask;

printDouble(mask);

cout << "Shift mask" << endl;

if (side == 'r') // Cycling Shift

n = (n >> amount) | (n << (length - amount));

else

n = (n << amount) | (n >> (length - amount));

number = (number & ~mask) | (n & mask); // Applying changes

}

void doubleTask() {

/\*

Вместо unsigned long long можно было бы использовать unsigned char[],

но это невозможно, т.к. в дополнительном задании требуется сдвиг во всём числе, а не только

в одном сегменте числа

\*/

union {

double number = 0.0;

unsigned long long repr;

};

cout << "Input a double: ";

number = readValue<double>();

printDouble(repr);

cout << " (" << number << ")" << endl;

while (true) {

char side;

int amount;

int begin;

int length;

do {

cout << "Enter shift side, (l)eft, (r)ight or (n)one: ";

side = readValue<char>();

} while (side != 'l' && side != 'r' && side != 'n');

if (side == 'n') return;

do {

cout << "Enter shift amount (-1 to cancel): ";

amount = readValue<int>();

} while (amount < -1);

if (amount == -1) return;

do {

cout << "Enter segment begin (-1 to cancel): ";

begin = readValue<int>();

} while (begin < -1);

if (begin == -1) return;

do {

cout << "Enter segment length (-1 to cancel): ";

length = readValue<int>();

} while (length < -1);

if (length == -1) return;

if (begin >= sizeof(long long) \* 8) return;

if (begin + length > sizeof(long long) \* 8) return;

printDouble(repr);

cout << endl;

varyDouble(repr, side, amount, begin, length);

printDouble(repr);

cout << "(" << number << ")" << endl;

}

}