

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| УГНС | | 09.00.00 | Информатика и вычислительная техника | | |
| Направление подготовки | | 09.03.01 | Информатика и вычислительная техника | | |
| Направленность (профиль) | |  | Системы автоматизированного проектирования | | |
| Форма обучения | |  | очная | | |
|  | |  |  | | |
| Факультет | |  | Информационных технологий и управления | | |
| Кафедра | |  | Систем автоматизированного проектирования и управления | | |
| Учебная дисциплина | |  | Вычислительные системы, сети и телекоммуникации | | |
| Курс | II | | | Группа | 423 |

Отчёт по контрольной работе №2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Исполнители: |  |  |  |  |
| обучающиеся группы 423 |  |  |  | Агишев Даниил Рустамович  Сафаров Рустам Фахриддинович  Ефремов Иван Андреевич |
|  |  | (дата, подпись) |  |  |
|  |  |  |  |  |
| Проверили: |  |  |  | Макарук Роман Валерьевич |
|  |  | (дата, подпись) |  | Федин Алексей Константинович |
|  |  |  |  |  |

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc158160395)

[2 Исходные данные 3](#_Toc158160396)

[3 Текст учебной программы 3](#_Toc158160397)

[4 Выводы по заданию 5](#_Toc158160398)

## 1 Постановка задачи

Разработать asm-вставку для учебной программы для проверки простоты числа.

## 2 Исходные данные

Учебная программа – определение кратности положительного целого числа пяти:

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

unsigned int number = 0;

int isNumberDivisibleByFive = 0;

const int five = 5;

cout << "N = ";

cin >> number;

\_asm {

XOR EDX, EDX

MOV EAX, number

DIV five

CMP EDX, 0

JNE L01

MOV isNumberDivisibleByFive, 1

L01:

}

cout << number;

if (isNumberDivisibleByFive) {

cout << " - делится на 5 без остатка" << endl;

}

else {

cout << " - делится на 5 с остатком" << endl;

}

system("pause");

}

## 3 Текст учебной программы

Исходный код программного решения:

#include <iostream>

#include<windows.h>

using namespace std;

enum class MENU { check = 1, exit = 2 };

template<typename T>

T checkInput() {

T userInput{};

while (!(cin >> userInput)) {

cin.clear();

cin.ignore(INT\_MAX, '\n');

cout << "Введены некорректные данные. Попробуйте снова." << endl;

}

cin.ignore(INT\_MAX, '\n');

return userInput;

}

int GetUnsInt() {

int input = checkInput<int>();

if (input < 0) {

cout << "Взято значение по модулю." << endl;

input = abs(input);

}

return input;

}

bool numIsPrime(unsigned int number) {

int isNumberDivisible = 0;

bool simple = true;

for (unsigned int i = 2; i <= sqrt(number); i++) {

const int denom = i;

\_asm {

XOR EDX, EDX

MOV EAX, number

DIV denom

CMP EDX, 0

JNE L01

MOV isNumberDivisible, 1

L01:

}

if (isNumberDivisible) {

simple = false;

break;

}

}

if (number <= 1)

simple = false;

return simple;

}

void SetColor(int text, int background)

{

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

SetConsoleTextAttribute(hConsoleHandle, (WORD)((background << 4) | text));

}

int main() {

setlocale(LC\_CTYPE, "Russian");

unsigned int number;

cout << "Лабораторная работа №2 по ВССиТ" << endl;

cout << "Выполнено студентами 423 группы Сафаровым Рустамом, Агишевым Даниилом, Ефремовым Иваном" << endl;

while (true) {

cout << "==============================================" << endl;

cout << "Выберите действие: " << endl;

cout << "1 - проверка простоты числа" << endl;

cout << "2 - завершение работы" << endl;

MENU choice = static\_cast<MENU>(checkInput<int>());

switch (choice) {

case MENU::check:

cout << "Введите число для выполнения проверки, является ли оно простым: ";

number = GetUnsInt();

if (numIsPrime(number)) {

SetColor(2, 0);

cout << endl << "Число " << number << " является простым." << endl << endl;

}

else {

SetColor(4, 0);

cout << endl << "Число " << number << " не является простым." << endl << endl;

}

SetColor(15, 0);

break;

case MENU::exit:

cout << "Работа программы была завершена пользователем." << endl;

return 0;

break;

default:

cout << "Введены некорректные данные. Попробуйте снова." << endl;

}

}

}

Далее представлены фрагменты ассемблер кода для каждой арифметической операции.

Таблица 1 – Ассемблерный код

|  |  |
| --- | --- |
| **Команда ассемблера** | **Комментарий** |
| MOV EAX, number | Перемещаем первый операнд в регистр EAX |
| XOR EDX, EDX | Обнуление значения в регистре EDX |
| DIV denom | Деление значения в регистре EAX на denom |
| CMP EDX, 0 | Сравнение значения в регистре EDX и 1 |
| JGE L01 | Если EDX больше или равен 1, переход на L01 |
| MOV isNumberDivisible, 1 | Перемещаем 1 в isNumberDivisible |
| L01: | Пустая метка |

## 4 Выводы по заданию

В ходе выполнения задания были изучены методы интеграции asm-вставок в код проекта и получены практические навыки в работе с задачей поиска простого числа.