Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное   
образовательное учреждение высшего образования   
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

Институт информатики, математики и электроники

Факультет информатики

Кафедра технической кибернетики

Отчет по лабораторной работе №2

Дисциплина: «Системное программирование и архитектура ЭВМ»

Выполнил: Абрамов Д.О.

Группа: 6208-010302D

Самара, 2021

**Задание на лабораторную работу**

1. В программе, разработанной в лабораторной работе 1 необходимо добавить код на ассемблере реализующий:
   1. Проверку делителя на равенство 0 — для устранения системной ошибки деление на ноль.
   2. Проверку делимого на переполнения частного — для устранения системной ошибки «частное велико».
   3. Проверку результата на переполнение.
2. В качестве комментария к каждой строке на ассемблере необходимо указать, какой промежуточный результат формируется в каком регистре.

1 вариант: (5\*c + 6\*a – b\*2) / (c + a/d)

**Код программы**

#include <iostream>

using namespace std;

// (5\*c + 6\*a – b\*2)/(c + a/d);

int main()

{

int a, b, c, d, null = 0, of = 0, result;

cout << "Enter first num ";

cin >> a;

cout << endl << "Enter second num ";

cin >> b;

cout << endl << "Enter third num ";

cin >> c;

cout << endl << "Enter four num ";

cin >> d;

\_\_asm

{

mov ecx, c; ecx = c

jo error\_of; ошибка переполнения

imul ecx, 5; ecx = 5 \* c

mov eax, a; eax = a

jo error\_of; ошибка переполнения

imul eax, 6; eax = 6 \* a

mov ebx, b; ebx = b

jo error\_of; ошибка переполнения

imul ebx, 2; ebx = b \* 2

neg ebx; ebx = -(b \* 2)

jo error\_of; ошибка переполнения

add ecx, eax; ecx = 5 \* c + 6 \* a

jo error\_of; ошибка переполнения

add ecx, ebx; ecx = 5 \* c + 6 \* a – b \* 2

push ecx; запихали 5 \* c + 6 \* a – b \* 2 в стек

mov eax, a; eax = a

mov ebx, d; ebx = d

cmp ebx, 0; ebx = 0 ?

cdq

jo error\_of; ошибка переполнения

je division\_by\_zero; ошибка деления на ноль

idiv ebx; eax = a / d

mov edx, eax; edx = eax = a / d

mov ecx, c; ecx = c

add ecx, edx; ecx = c + a / d

jo error\_of; ошибка переполнения

pop eax; eax = 5 \* c + 6 \* a – b \* 2

cdq

jo error\_of; ошибка переполнения

je division\_by\_zero; ошибка деления на ноль

idiv ecx; eax = (5 \* c + 6 \* a – b \* 2) (c + a / d)

mov result, eax;

jmp e1;

division\_by\_zero:; флаг ошибки деления на ноль

mov edx, null; edx = zero

inc edx;

mov null, edx;

e1:

jmp e2;

error\_of:; флаг ошибки переполнения

mov edx, of; edx = of

inc edx;

mov of, edx;

e2:

}

if (of == 1)

cout << "\nOut of size!\n";

if (null == 1)

cout << "\nDivision by zero!\n";

if (of == 0 && null == 0)

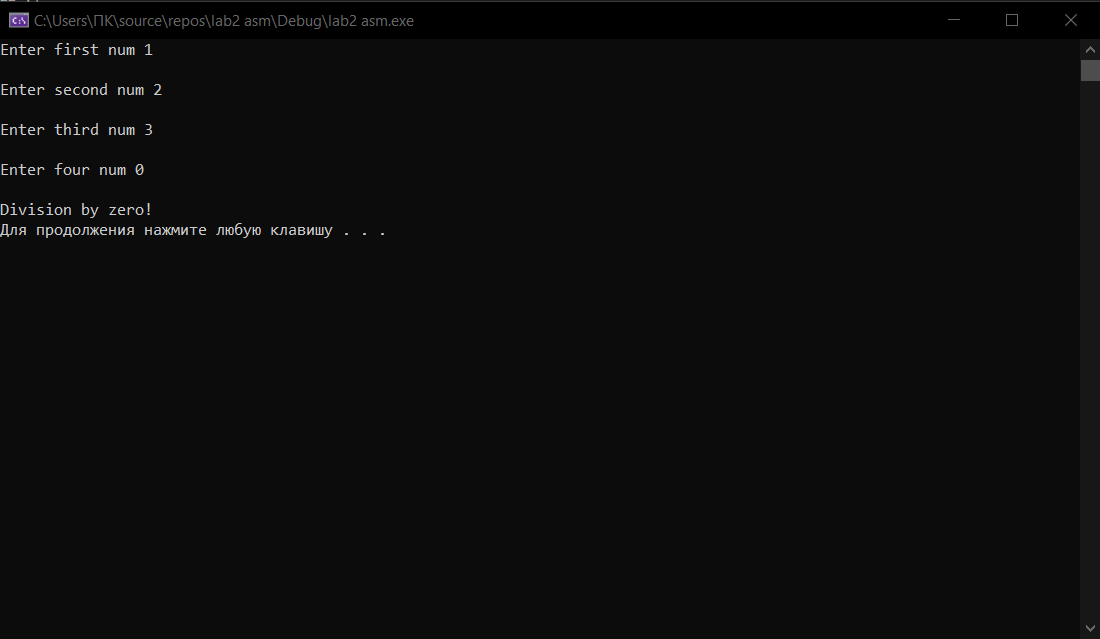
cout << "\nResult asm result = " << result;

if (d != 0 && (c + a / d) != 0 && null == 0 && of == 0)

cout << "\nResult c++ = " << (5 \* c + 6 \* a - b \* 2) / (c + a / d) << endl;

system("pause");

}

****

