*Федеральное государственное бюджетное образовательное учреждение высшего образования*

*«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)*

---

ФАКУЛЬТЕТ _____ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ_____

КАФЕДРА _____КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ_____

**О Т Ч Е Т**

**по лабораторной работе № ___2__**

**Дисциплина: __Машинно-зависимые языки и основы компиляции_____**

**Название лабораторной работы:** _Изучение среды и отладчика_ассемблера

_____

Студент гр. ___**ИУ6-42Б** _____ **_Медведев АЕ_**
(Подпись, дата) (И.О. Фамилия)

Преподаватель _____ _____
(Подпись, дата) (И.О. Фамилия)

Москва, 2021_

**Задание**

1. Разработать программу, вычисляющую заданное выражение. Просмотреть в отладчике и зафиксировать в отчете ход выполнения вычислений (покомандно). Убедиться в правильности программы.

**Лабораторная работ №2. Программирование целочисленных вычислений.**

Вычислить целочисленное выражение:

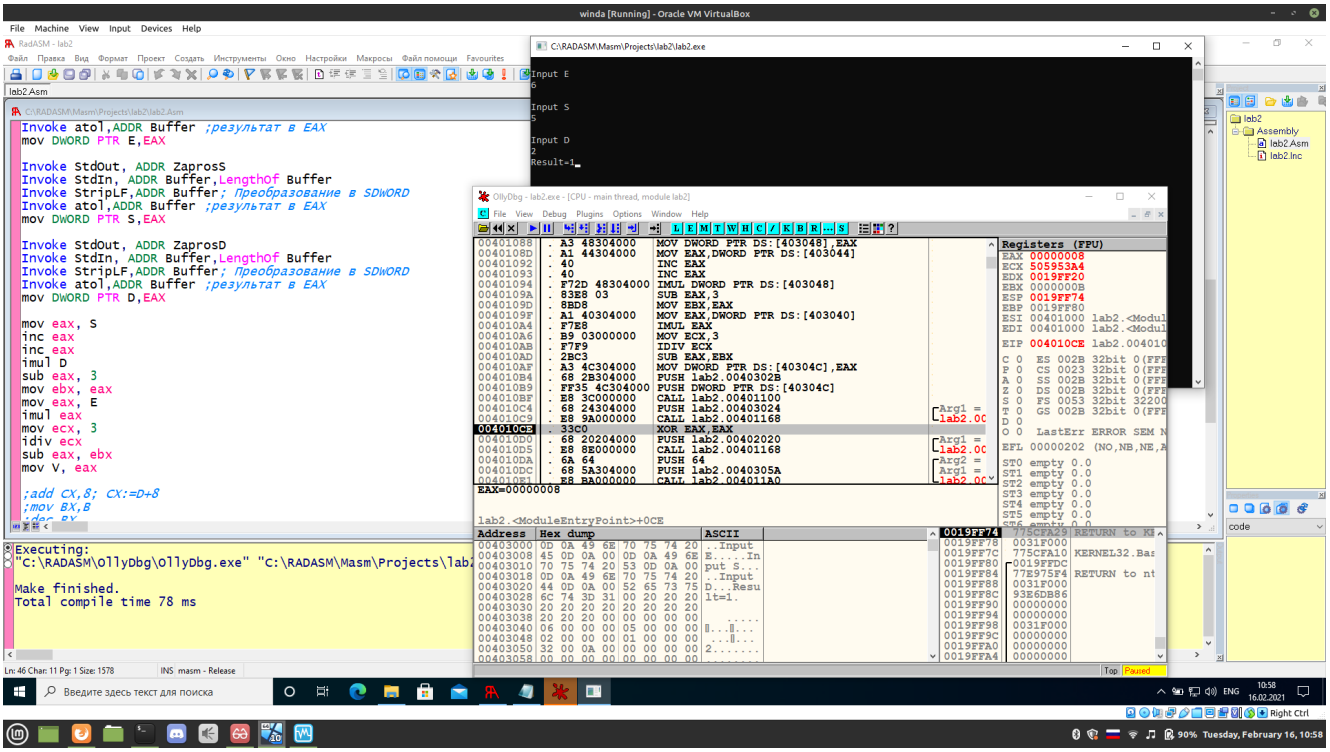
$$v = \frac{e^2}{3} - (s+2) * d + 3$$

**Код программы:**

```
.586
.MODEL flat, stdcall
OPTION CASEMAP:NONE
Include kernel32.inc
Include masm32.inc
IncludeLib kernel32.lib
IncludeLib masm32.lib
.CONST
MsgExit DB 13,10,"Press Enter to Exit",0AH,0DH,0
.DATA
ZaprosE DB 13,10,'Input E',13,10,0
ZaprosS DB 13,10,'Input S',13,10,0
ZaprosD DB 13,10,'Input D',13,10,0
Result DB 'Result='
ResStr DB 16 DUP (' '),0

.DATA?
E DWORD ?
S DWORD ?
D DWORD ?
V DWORD ?
Buffer DB 10 DUP (?)
inbuf DB 100 DUP (?)

.CODE
Start:
```

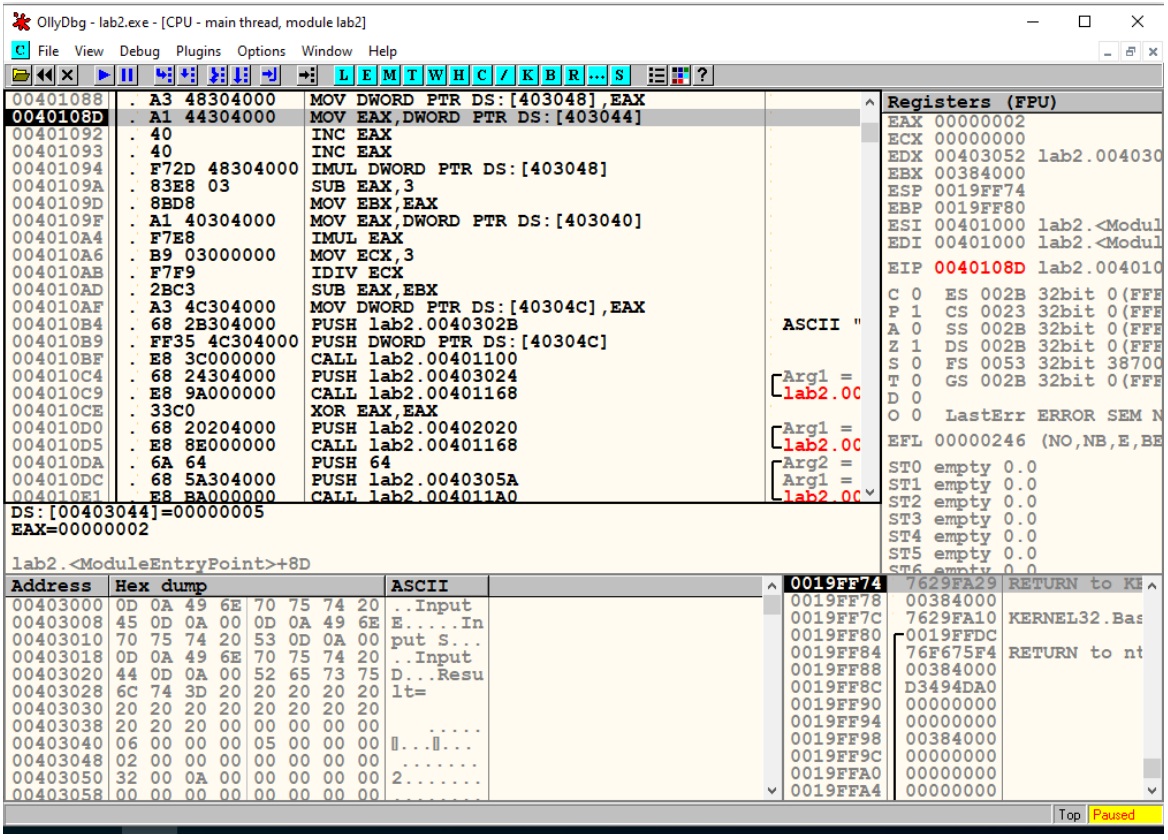
```
Invoke StdOut, ADDR ZaprosE
Invoke StdIn, ADDR Buffer,LengthOf Buffer
Invoke StripLF,ADDR Buffer
Invoke atol,ADDR Buffer
mov DWORD PTR E,EAX

Invoke StdOut, ADDR ZaprosS
Invoke StdIn, ADDR Buffer,LengthOf Buffer
Invoke StripLF,ADDR Buffer
Invoke atol,ADDR Buffer
mov DWORD PTR S,EAX

Invoke StdOut, ADDR ZaprosD
Invoke StdIn, ADDR Buffer,LengthOf Buffer
Invoke StripLF,ADDR Buffer
Invoke atol,ADDR Buffer
mov DWORD PTR D,EAX

mov eax, S
inc eax
inc eax
imul D
sub eax, 3
mov ebx, eax
mov eax, E
imul eax
mov ecx, 3
idiv ecx
sub eax, ebx
mov V, eax

;add CX,8; CX:=D+8
;mov BX,B
;dec BX
;mov AX,A
;add AX,D; AX:=A+D
;imul BX ; DX:AX:=(A+D)*(B-1)
;idiv CX ; AX:=(DX:AX):CX
;mov V,AX;

Invoke dwtoa,V,ADDR ResStr;
Invoke StdOut,ADDR Result


XOR EAX,EAX
Invoke StdOut,ADDR MsgExit
```

```
Invoke StdIn,ADDR inbuf,LengthOf inbuf
Invoke ExitProcess,0
End Start
```

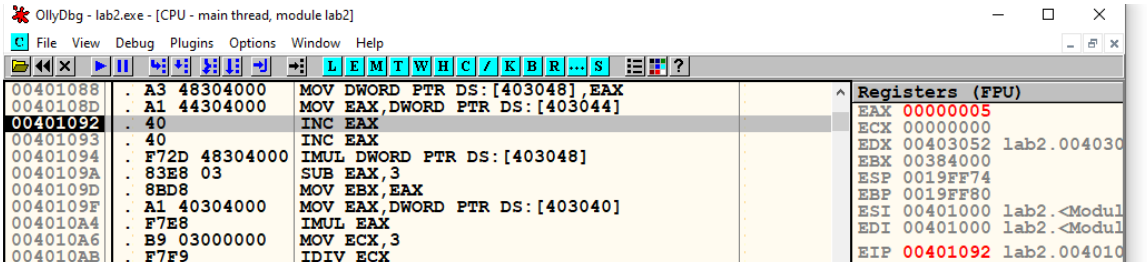



(рис. 1 код)

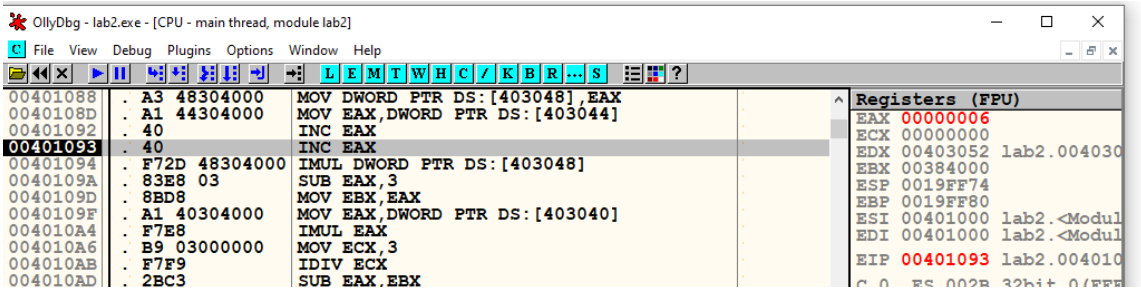
Покомандно:



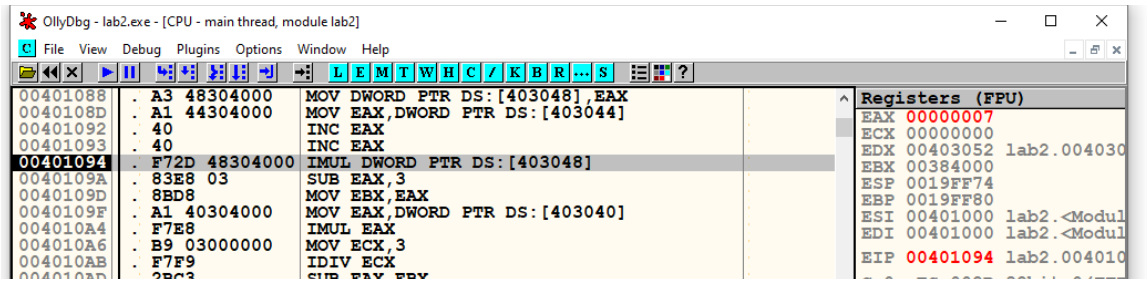

(рис. 2 считать вторую переменную в регистр)



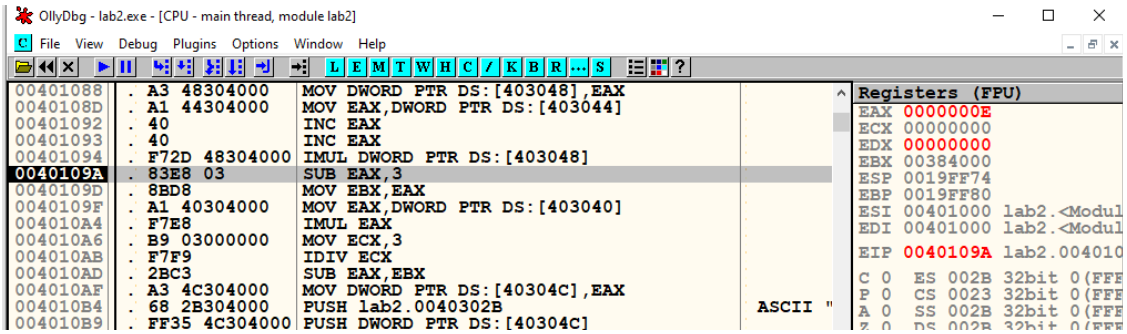

(рис. 3 увеличить вторую переменную на 1)



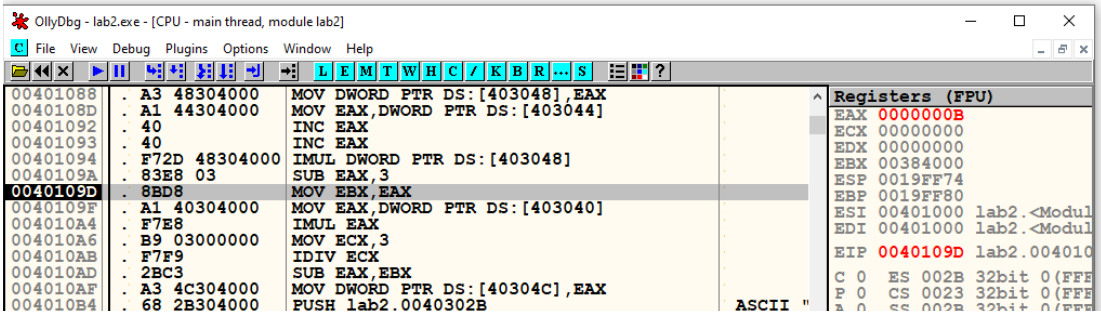

(рис. 4 увеличить вторую переменную на 1)



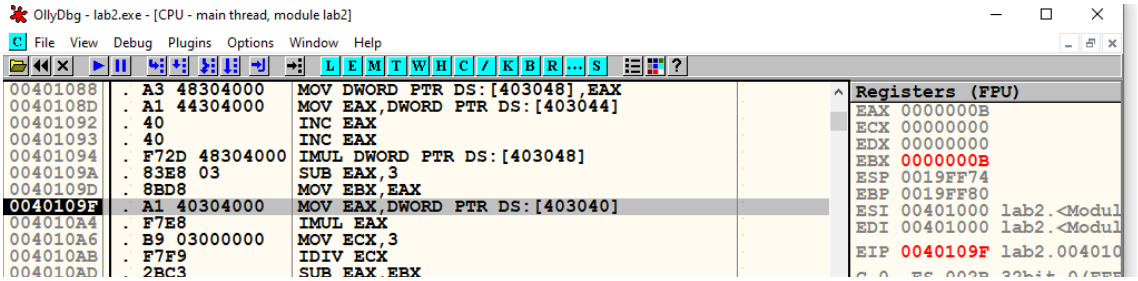

(рис. 5 умножить вторую переменную на 3 переменную)



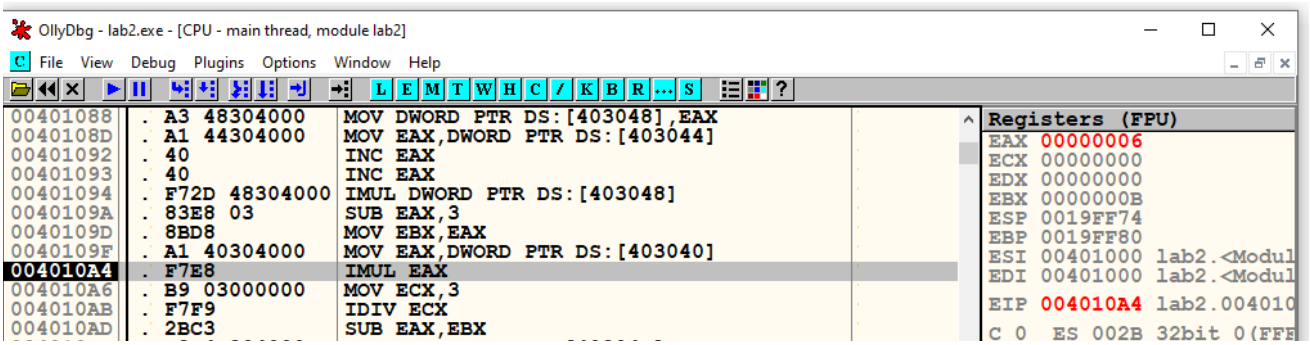

(рис. 6 вычесть из произведения 3)



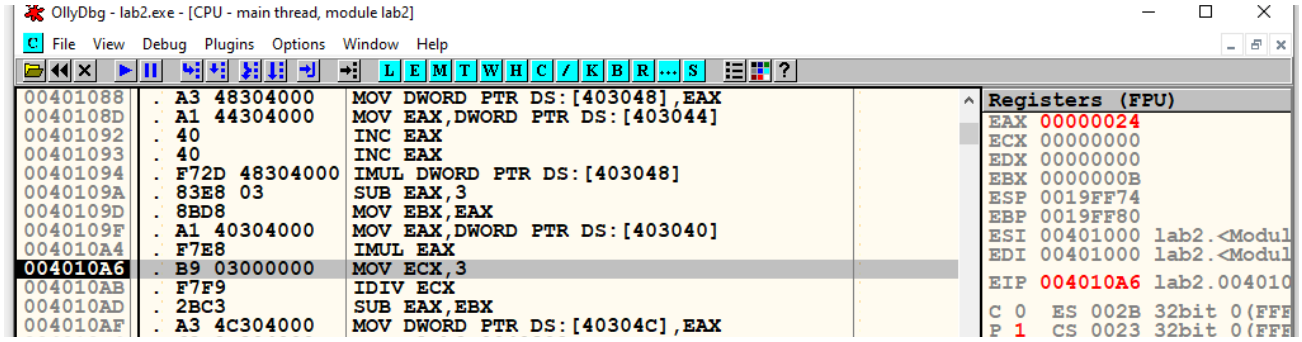

(рис. 7 записать результат в регистр EBX)



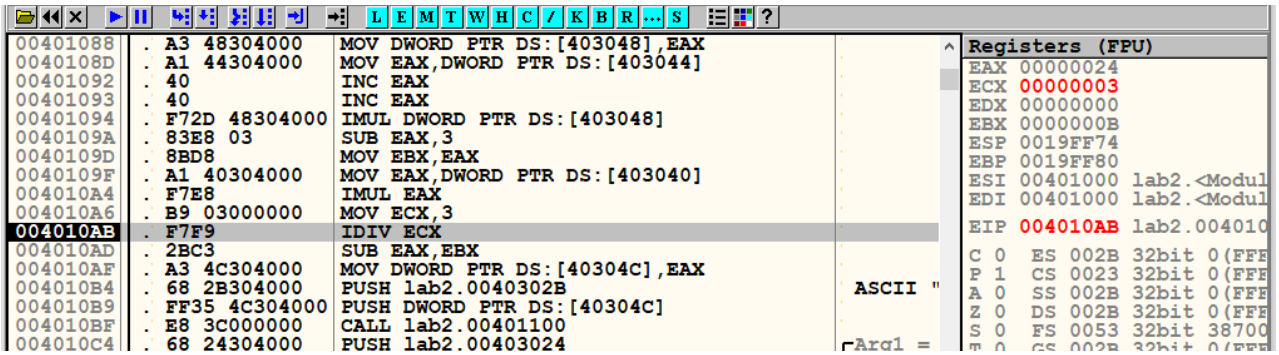

(рис. 8 загрузить в переменную EAX первую переменную)



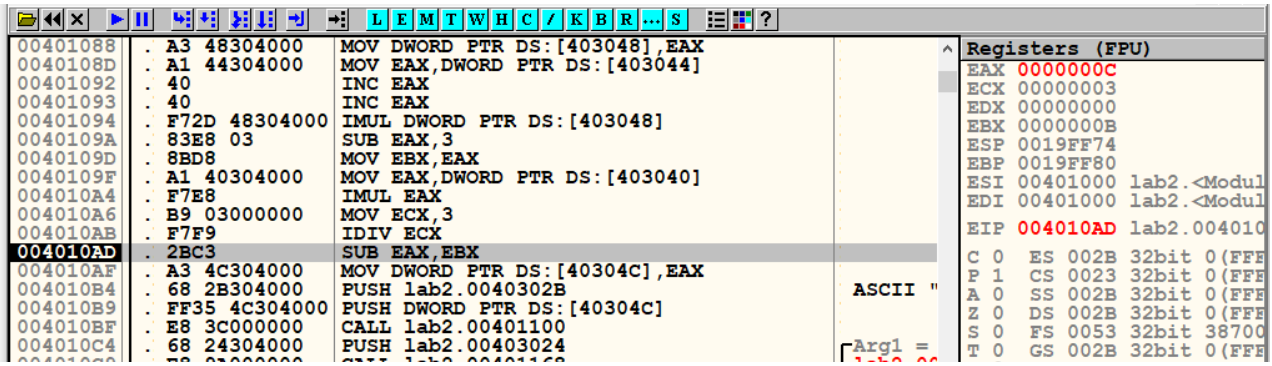

(рис. 9 умножить первую переменную на себя)



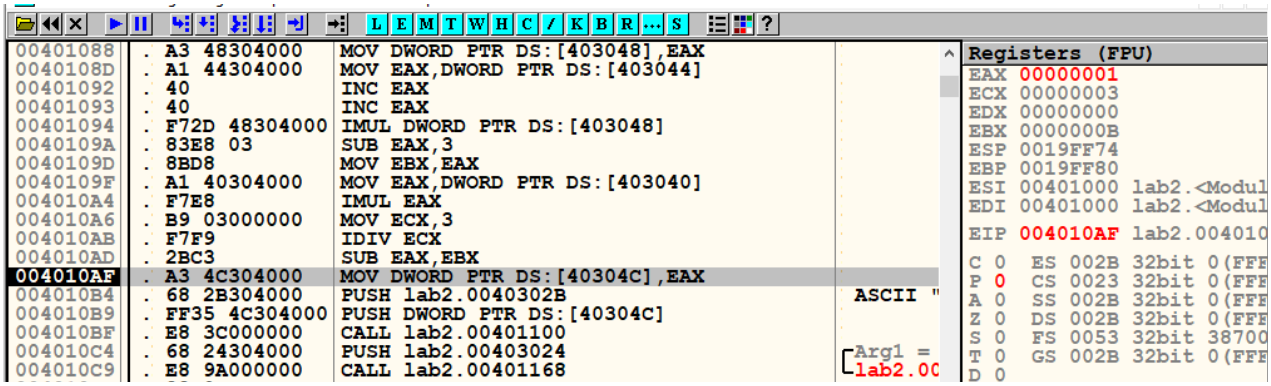

(рис. 10 загрузить в ECX 3)



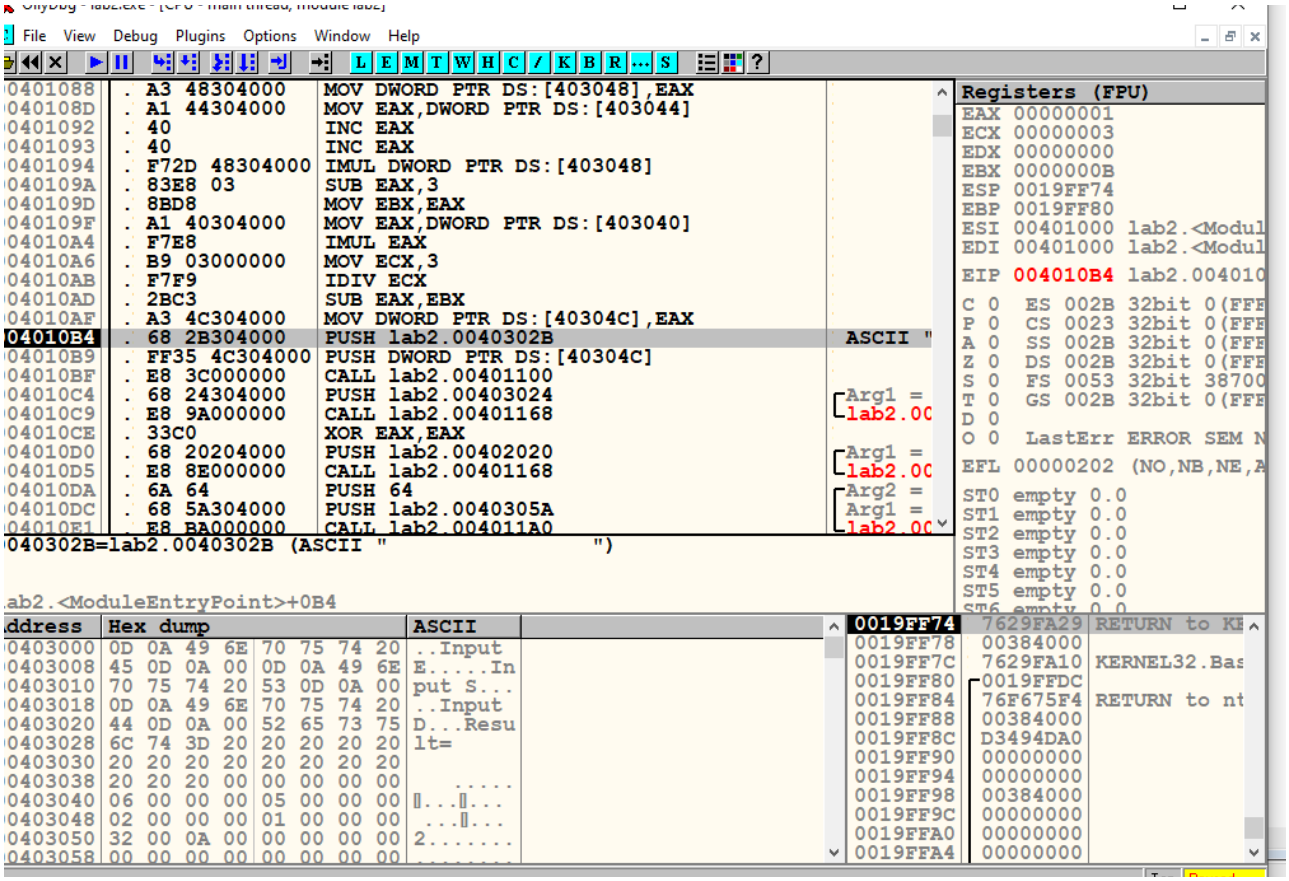

(рис. 11 разделить квадрат первой переменной на 3)




(рис. 12 вычесть два найденных значения)




(рис. 13 загрузить ответ в переменную V)




(рис. 14 вывод на экран)

| Ввод e | Ввод s | Ввод d | Ожидалось | Выполнение программы |
|---|---|---|---|---|
| 1 | 1 | 1 | 0.333333334 | 0 |
| -5 | 1 | 1 | 8.333333334 | 8 |
| -5 | 1 | 0 | 11.333333332 | 11 |
| -8 | 7 | 1 | 15.333333334 | 15 |
| 0 | 0 | 0 | 3 | 3 |

2. Посмотреть в отладчике форматы 3-4 команд mov и расшифровать двоичные коды этих команд, используя материалы теоретической части.

Команда : mov ebx, eax
Код : 8bd8
Двоичный код: 10001011 11 011 000

Команда : mov ecx, 3
Код : b9 03000000
Двоичный код: 10111 001 00000011

Команда : mov eax, dword ptr ds:[403044]
Код : a1 44304000
Двоичный код: 1010001 01000100 00110000 0100000 00000000

Контрольные вопросы.

1. Что такое машинная команда? Какие форматы имеют машинные команды процессора IA32? Чем различаются эти форматы?

Это элементарная инструкция компьютеру. Машинная команда состоит из двух частей: операционной и адресной. Операционная часть команды – это группа разрядов в команде, предназначенная для указания кода операции.

2. Назовите мнемоники основных команд целочисленной арифметики. Какие форматы для них можно использовать?
- Перемещение
mov reg, reg
mov mem, reg
mov reg, mem
- Работа со стеком
PUSH imm16 / imm32 / r16 / r32 / m16 / m32
POP r16 / r32 / m16 / m32
- инкремент
INC reg/mem

3. Сформулируйте основные правила построения линейной программы вычисления заданного выражения.
Все операции в программе выполняются одна за другой(нет распараллеливания).

4. Почему ввод-вывод на языке ассемблера не программируют с ис-

пользованием соответствующих машинных команд? Какая библиотека используется для организации ввода вывода в данной лабораторной?

Через машинные команды программировать вводи и вывод данных в поток и из потока соответственно очень сложно.
В данной лабораторной используются команды ввода вывода стандартной библиотеки среды RADASM32.

5. Расскажите, какие процедуры использую для организации ввода вывода. Какие операции выполняет каждая процедура?

- Ввод:
StdIn PROC lpszBuffer:DWORD, bLen:DWORD
- Добавление символа конца строки:
StripLF PROC lpszBuffer:DWORD
- Преобразование строки в число:
atol proc lpszBuffer:DWORD
- Вывод строки:
StdOut PROC lpszBuffer:DWORD ; буфер вывода, зав. Нулем
- Преобразование числа в строку символов:
dwtoa PROC public dwValue:DWORD, lpBuffer:PTR BYTE

**Вывод:** В ходе лабораторной работы, была разработана программа вычисляющая математическое выражение на языке ассемблера, изучена работа дебагера RadAsm.