



Министерство науки и высшего образования Российской  
Федерации  
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 \_\_\_\_\_  
НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Компьютерные системы и сети.

**О Т Ч Е Т**

по лабораторной работе № \_\_\_\_2\_\_\_\_

Дисциплина: Машинно-зависимые языки и основы компиляции

Название лабораторной работы: Программирование целочисленных  
вычислений

Студент гр. ИУ6-42Б \_\_\_\_\_ Медведев АЕ  
(Подпись, дата) (И.О. Фамилия)

Преподаватель \_\_\_\_\_  
(Подпись, дата) (И.О. Фамилия)

Москва, 2021

**Задание (Вариант 19):**

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

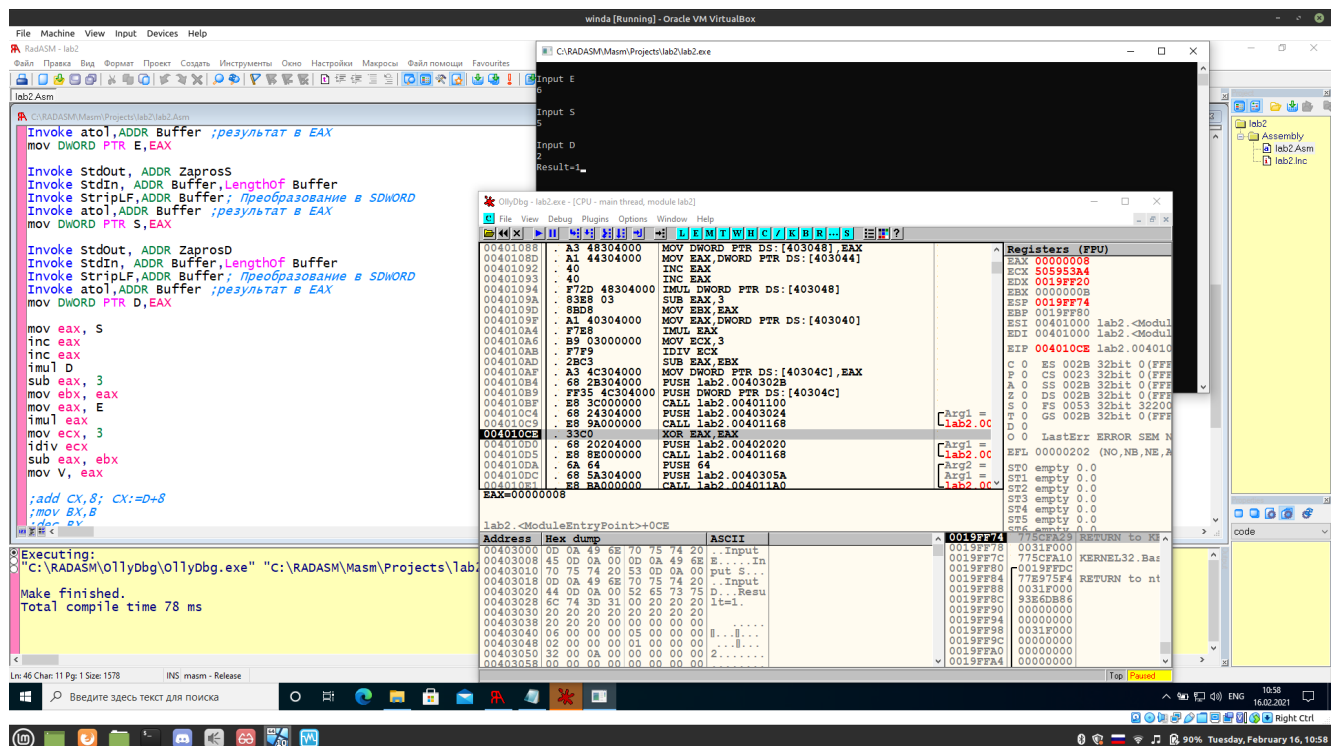


Рисунок 1 — код программы

По командно:

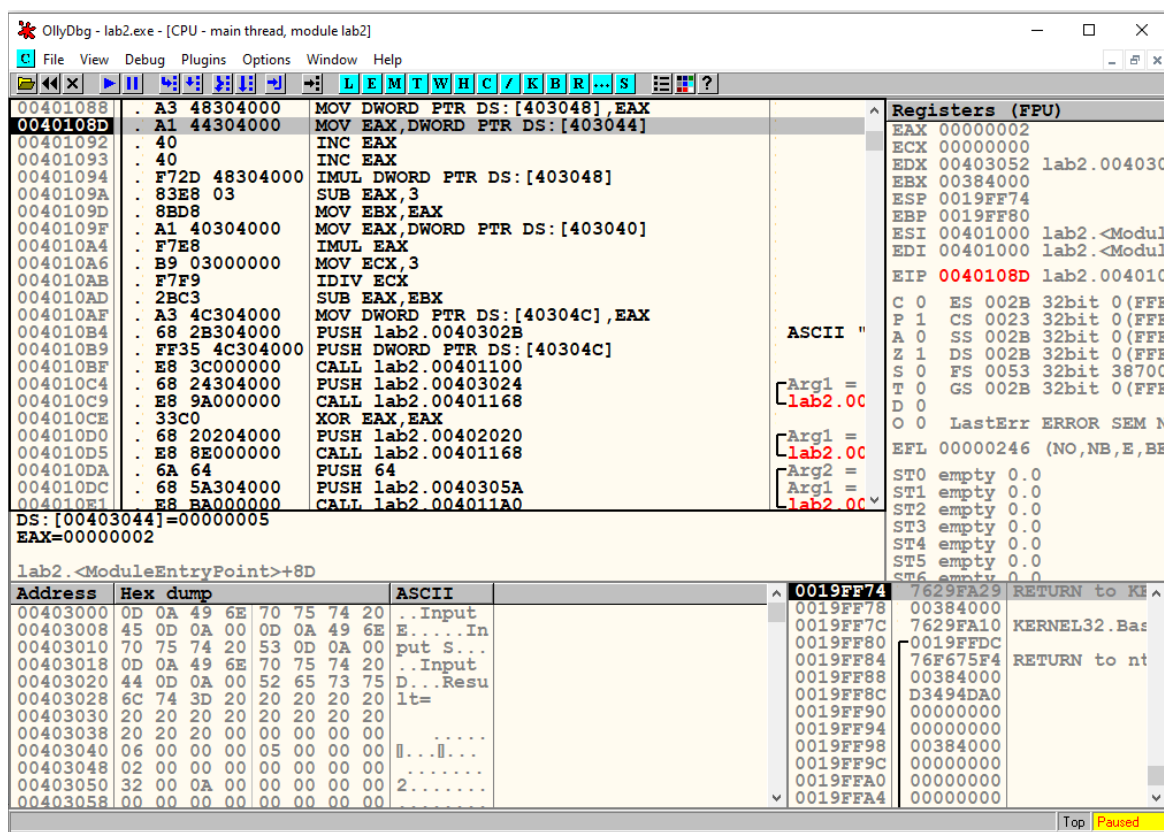


Рисунок 2 - считать вторую переменную в регистр

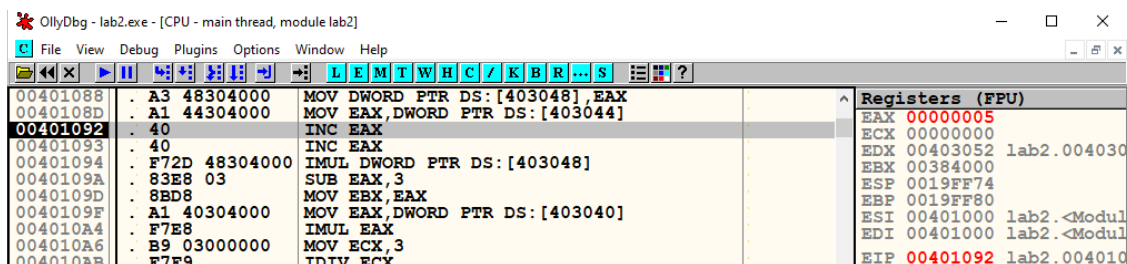


Рисунок 3 - увеличить вторую переменную на 1

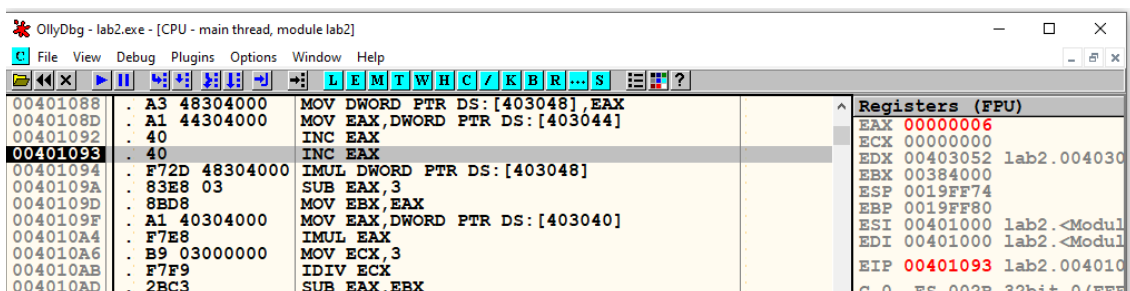


Рисунок 4 - увеличить вторую переменную на 1

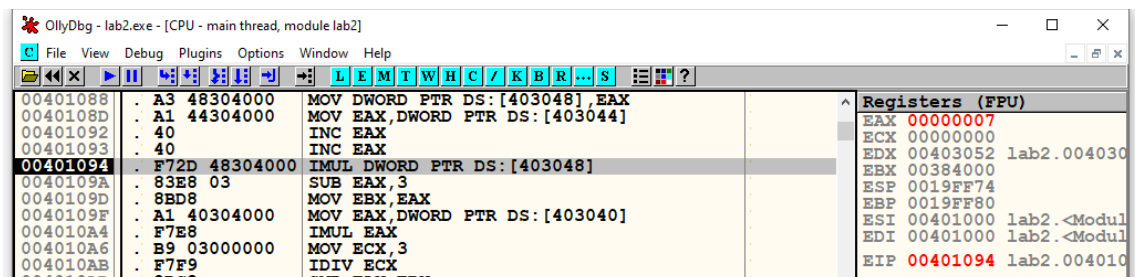


Рисунок 5 - умножить вторую переменную на 3 переменную

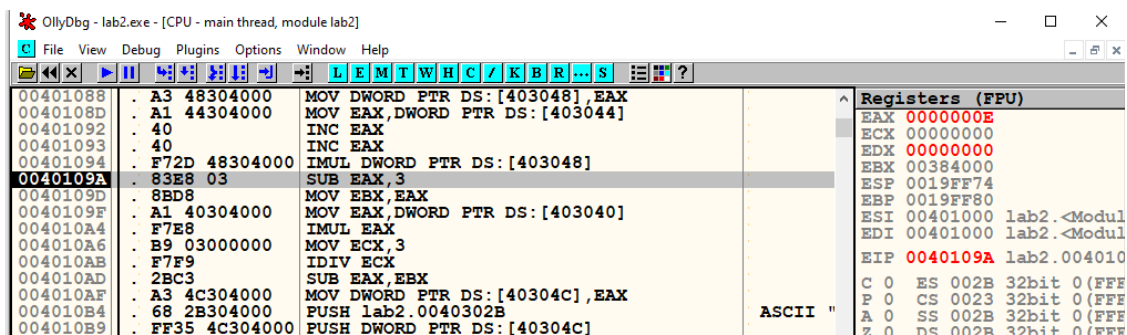


Рисунок 6 - вычесть из произведения 3

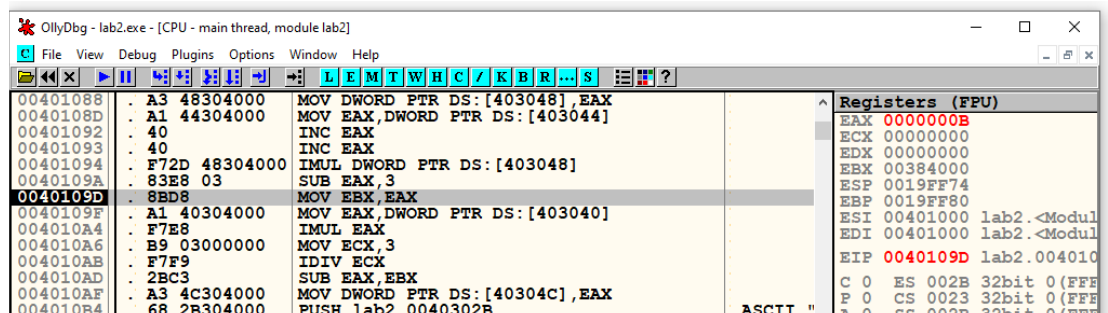


Рисунок 7 - записать результат в регистр EBX

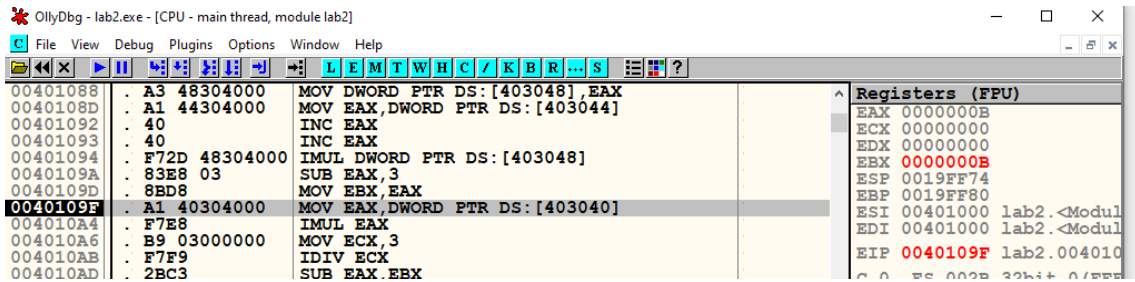


Рисунок 8 - загрузить в переменную EAX первую переменную

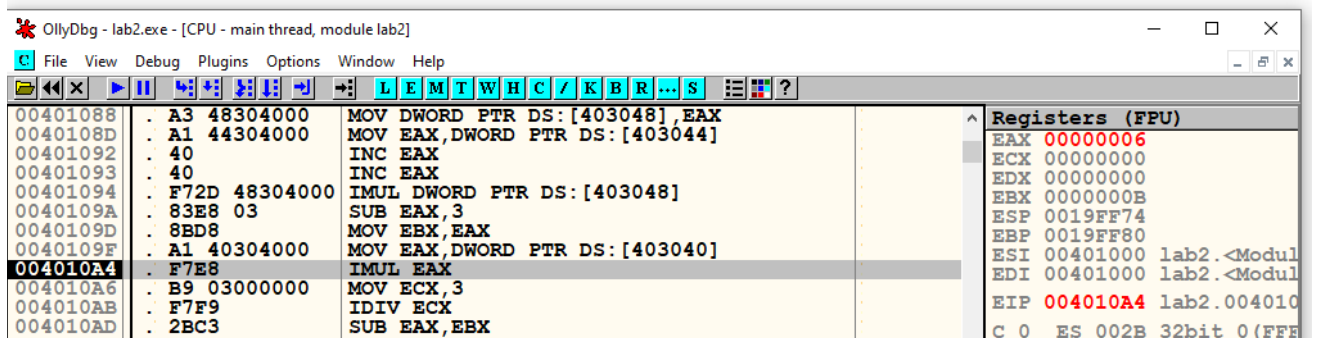


Рисунок 9 - умножить первую переменную на себя

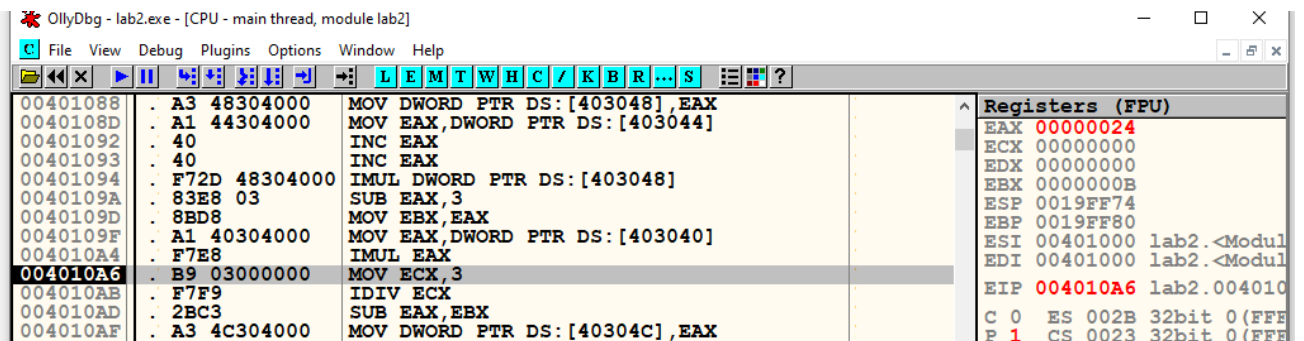


Рисунок 10 - загрузить в ECX 3

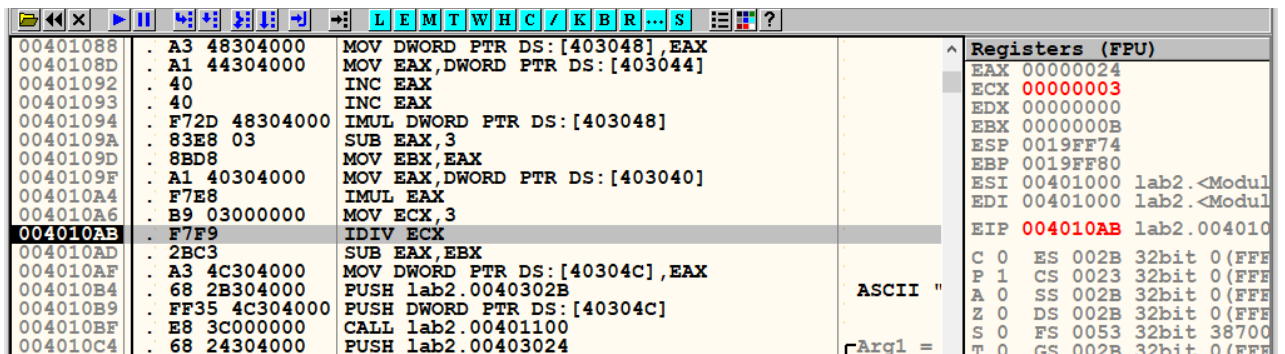


Рисунок 11 - разделить квадрат первой переменной на 3



00401088	. A3 48304000	MOV DWORD PTR DS:[403048],EAX		Registers (FPU)
0040108D	. A1 44304000	MOV EAX,DWORD PTR DS:[403044]		EAX 0000000C
00401092	. 40	INC EAX		ECX 00000003
00401093	. 40	INC EAX		EDX 00000000
00401094	. F72D 48304000	IMUL DWORD PTR DS:[403048]		EBX 0000000B
0040109A	. 83E8 03	SUB EAX,3		ESP 0019FF74
0040109D	. 8BD8	MOV EBX,EAX		EBP 0019FF80
0040109F	. A1 40304000	MOV EAX,DWORD PTR DS:[403040]		ESI 00401000 lab2.<Modul
004010A4	. F7E8	IMUL EAX		EDI 00401000 lab2.<Modul
004010A6	. B9 03000000	MOV ECX,3		EIP 004010AD lab2.004010
004010AB	. F7F9	IDIV ECX		C 0 ES 002B 32bit 0 (FFF
004010AD	. 2BC3	SUB EAX,EBX		P 1 CS 0023 32bit 0 (FFF
004010AF	. A3 4C304000	MOV DWORD PTR DS:[40304C],EAX		A 0 SS 002B 32bit 0 (FFF
004010B4	. 68 2B304000	PUSH lab2.0040302B	ASCII "	Z 0 DS 002B 32bit 0 (FFF
004010B9	. FF35 4C304000	PUSH DWORD PTR DS:[40304C]		S 0 FS 0053 32bit 38700
004010BF	. E8 3C000000	CALL lab2.00401100		T 0 GS 002B 32bit 0 (FFF
004010C4	. 68 24304000	PUSH lab2.00403024	[Arg1 = 1ab2.0c	
004010C9	. E8 9A000000	CALL lab2.00401168		

Рисунок 12 - вычесть два найденных значения

00401088	. A3 48304000	MOV DWORD PTR DS:[403048],EAX		Registers (FPU)
0040108D	. A1 44304000	MOV EAX,DWORD PTR DS:[403044]		EAX 00000001
00401092	. 40	INC EAX		ECX 00000003
00401093	. 40	INC EAX		EDX 00000000
00401094	. F72D 48304000	IMUL DWORD PTR DS:[403048]		EBX 0000000B
0040109A	. 83E8 03	SUB EAX,3		ESP 0019FF74
0040109D	. 8BD8	MOV EBX,EAX		EBP 0019FF80
0040109F	. A1 40304000	MOV EAX,DWORD PTR DS:[403040]		ESI 00401000 lab2.<Modul
004010A4	. F7E8	IMUL EAX		EDI 00401000 lab2.<Modul
004010A6	. B9 03000000	MOV ECX,3		EIP 004010AF lab2.004010
004010AB	. F7F9	IDIV ECX		C 0 ES 002B 32bit 0 (FFF
004010AD	. 2BC3	SUB EAX,EBX		P 0 CS 0023 32bit 0 (FFF
004010AF	. A3 4C304000	MOV DWORD PTR DS:[40304C],EAX		A 0 SS 002B 32bit 0 (FFF
004010B4	. 68 2B304000	PUSH lab2.0040302B	ASCII "	Z 0 DS 002B 32bit 0 (FFF
004010B9	. FF35 4C304000	PUSH DWORD PTR DS:[40304C]		S 0 FS 0053 32bit 38700
004010BF	. E8 3C000000	CALL lab2.00401100		T 0 GS 002B 32bit 0 (FFF
004010C4	. 68 24304000	PUSH lab2.00403024	[Arg1 = 1ab2.0c	
004010C9	. E8 9A000000	CALL lab2.00401168		D 0

Рисунок 13 - загрузить ответ в переменную V

00401088	. A3 48304000	MOV DWORD PTR DS:[403048],EAX		Registers (FPU)
0040108D	. A1 44304000	MOV EAX,DWORD PTR DS:[403044]		EAX 00000001
00401092	. 40	INC EAX		ECX 00000003
00401093	. 40	INC EAX		EDX 00000000
00401094	. F72D 48304000	IMUL DWORD PTR DS:[403048]		EBX 0000000B
0040109A	. 83E8 03	SUB EAX,3		ESP 0019FF74
0040109D	. 8BD8	MOV EBX,EAX		EBP 0019FF80
0040109F	. A1 40304000	MOV EAX,DWORD PTR DS:[403040]		ESI 00401000 lab2.<Modul
004010A4	. F7E8	IMUL EAX		EDI 00401000 lab2.<Modul
004010A6	. B9 03000000	MOV ECX,3		EIP 004010B4 lab2.004010
004010AB	. F7F9	IDIV ECX		C 0 ES 002B 32bit 0 (FFF
004010AD	. 2BC3	SUB EAX,EBX		P 0 CS 0023 32bit 0 (FFF
004010AF	. A3 4C304000	MOV DWORD PTR DS:[40304C],EAX		A 0 SS 002B 32bit 0 (FFF
004010B4	. 68 2B304000	PUSH lab2.0040302B	ASCII "	Z 0 DS 002B 32bit 0 (FFF
004010B9	. FF35 4C304000	PUSH DWORD PTR DS:[40304C]		S 0 FS 0053 32bit 38700
004010BF	. E8 3C000000	CALL lab2.00401100		T 0 GS 002B 32bit 0 (FFF
004010C4	. 68 24304000	PUSH lab2.00403024	[Arg1 = 1ab2.0c	
004010C9	. E8 9A000000	CALL lab2.00401168		D 0
004010CE	. 33C0	XOR EAX,EAX	[Arg2 = 1ab2.0c	
004010D0	. 68 20204000	PUSH lab2.00402020	[Arg1 = 1ab2.0c	
004010D5	. E8 8E000000	CALL lab2.00401168	[Arg1 = 1ab2.0c	
004010DA	. 6A 64	PUSH 64		
004010DC	. 68 5A304000	PUSH lab2.0040305A		
004010E1	. E8 BA000000	CALL lab2.004011A0		
0040302B=lab2.0040302B (ASCII " ")				
lab2.<ModuleEntryPoint>+0B4				
address	Hex dump	ASCII		
0403000	0D 0A 49 6E 70 75 74 20	..Input	0019FF74	7629FA29 RETURN to KE
0403008	45 0D 0A 00 0D 0A 49 6E	E....In	0019FF78	00384000
0403010	70 75 74 20 53 0D 0A 00	put S...	0019FF7C	7629FA10 KERNEL32.Bas
0403018	0D 0A 49 6E 70 75 74 20	..Input	0019FF80	0019FFDC
0403020	44 0D 0A 00 52 65 73 75	D...Resu	0019FF84	76F675F4 RETURN to nt
0403028	6C 74 3D 20 20 20 20 20	lt=	0019FF88	00384000
0403030	20 20 20 20 20 20 20 20		0019FF8C	D3494DA0
0403038	20 20 20 00 00 00 00 00	.....	0019FF90	00000000
0403040	06 00 00 00 05 00 00 00	l...ll...	0019FF94	00000000
0403048	02 00 00 00 01 00 00 00	...ll...	0019FF98	00384000
0403050	32 00 0A 00 00 00 00 00	2.....	0019FF9C	00000000
0403058	00 00 00 00 00 00 00 00		0019FFA0	00000000
			0019FFA4	00000000

Рисунок 14 - вывод на экран

Таблица тестов

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 использованием 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:
  - atoi 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.