



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 (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

**ОТЧЕТ**

по лабораторной работе № 2

Дисциплина: МЗЯиОК

Студент

ИУ6-43Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

В.К. Залыгин  
(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

\_\_\_\_\_  
(И.О. Фамилия)

Москва, 2024

## Цель работы

Изучение форматов машинных команд, команд целочисленной арифметики ассемблера и программирование целочисленных вычислений.

## Задание

Вычислить целочисленное выражение (смотреть рисунок 1).

$$d = a * x - 3 * \left(b + \frac{3}{k}\right)$$

Рисунок 1 – выражение

## Код программы

Ниже приведен листинг программы.

```
section .data
REQ_A  db  "Введите A:",10
REQ_A_LEN equ  $-REQ_A
REQ_X  db  "Введите X:",10
REQ_X_LEN equ  $-REQ_X
REQ_B  db  "Введите B:",10
REQ_B_LEN equ  $-REQ_B
REQ_K  db  "Введите K:",10
REQ_K_LEN equ  $-REQ_K
RES  db  "Результат вычисления выражения d = a * x - 3 * (b + 3 /
k):",10
RES_LEN  equ  $-RES
ERR  db  "Введены некорректные данные. Завершение работы.",10
ERR_LEN  equ  $-ERR
THREE  dd  3
section .bss
BUFFER  resb  10
```

```

BUFFER_LEN equ  $-BUFFER
A  resd  1
B  resd  1
D  resd  1
K  resd  1
X  resd  1
section .text
global _start ; d = a * x - 3 * (b + 3 / k)
_start:
    ; input the number A
    mov  rax,  1
    mov  rdi,  1
    mov  rsi,  REQ_A
    mov  rdx,  REQ_A_LEN
    syscall
    mov  rax,  0
    mov  rdi,  0
    mov  rsi,  BUFFER
    mov  rdx,  BUFFER_LEN
    syscall
    call StrToInt64
    cmp  rbx,  0
    jne  .err
    mov  [A],  eax
    ; input the number X
    mov  rax,  1
    mov  rdi,  1
    mov  rsi,  REQ_X

```

```

mov  rdx,  REQ_X_LEN
syscall

mov  rax,  0
mov  rdi,  0
mov  rsi,  BUFFER
mov  rdx,  BUFFER_LEN
syscall

call  StrToInt64
cmp  rbx,  0
jne  .err
mov  [X],  eax
; input the number B

mov  rax,  1
mov  rdi,  1
mov  rsi,  REQ_B
mov  rdx,  REQ_B_LEN
syscall

mov  rax,  0
mov  rdi,  0
mov  rsi,  BUFFER
mov  rdx,  BUFFER_LEN
syscall

call  StrToInt64
cmp  rbx,  0
jne  .err
mov  [B],  eax
; input the number K

mov  rax,  1

```

```

mov rdi, 1
mov rsi, REQ_K
mov rdx, REQ_K_LEN
syscall
mov rax, 0
mov rdi, 0
mov rsi, BUFFER
mov rdx, BUFFER_LEN
syscall
call StrToInt64
cmp rbx, 0
jne .err
cmp eax, 0
je .err
mov [K], eax
; compute d = a * x - 3 * (b + 3 / k)
xor rdx, rdx
mov eax, [A]
imul dword[X]
mov [A], eax
xor rdx, rdx
mov eax, 3
idiv dword[K]
add eax, [B]
imul dword[THREE]
mov [B], eax
mov eax, [A]
sub eax, [B]

```

```

    mov    [D],    eax
; output
    mov    rax,    1
    mov    rdi,    1
    mov    rsi,    RES
    mov    rdx,    RES_LEN
    syscall

    xor    rax,    rax
    mov    eax,    [D]
    mov    rsi,    BUFFER
    call   IntToStr64
    mov    rdx,    rax
    mov    rax,    1
    mov    rdi,    1
    syscall

; exit
    mov    rdi,    0
.exit:  mov    rax,    60
        syscall

.err:   mov    rax,    1
        mov    rdi,    1
        mov    rsi,    ERR
        mov    rdx,    ERR_LEN
        syscall
        mov    rdi,    1
        jmp    .exit
%include "../lib.asm"

```

Покомандное выполнение вычислений представлено на рисунках 2–13.



Рисунок 2 – выполнение вычислений



Рисунок 3 – выполнение вычислений



edb - /home/vzalygin/bmstu-ics6/asm/lab2/lab2.e [2878]

File View Debug Plugins Options Help

lab2.e: No Analysis Found

Registers

RAX	0000000000000000
RCX	0000000000000000
RDY	0000000000000000
RBX	0000000000000000
RSP	00007ffe6206b830
RBP	0000000000000000
RSI	0000000000000000
RDI	0000000000000000
R8	0000000000000000
R9	0000000000000000
R10	0000000000000000
R11	0000000000000000
R12	0000000000000000
R13	0000000000000000
R14	0000000000000000
R15	0000000000000000
RIP	0000000000000000 </home/vzalygin/bmstu-ics6/asm/lab2/lab2.e>

C 0 ES 0000  
P 1 CS 0033  
A 0 SS 002b  
Z 1 DS 0000  
S 0 FS 0000 (0000000000000000)  
T 0 GS 0000 (0000000000000000)  
D 0  
O 0  
EFL 00000246 (NO,AE,E,BE,NS,P,GE,LE)  
ST0 empty 0.0  
ST1 empty 0.0  
ST2 empty 0.0  
ST3 empty 0.0  
ST4 empty 0.0  
ST5 empty 0.0  
ST6 empty 0.0  
ST7 empty 0.0  
FTR ffff 3 2 1 0 E S P U C  
FSR 0000 Cond 0 0 0 0 Err 0 0 0 0  
FCR 037f Prec NEAR.64 Mask 1 1 1  
Last insn 0000000000000000

Bookmarks Registers

Data Dump

0x0000000000000000-0x0000000000000000

Stack

00007ffe:6206b830	0000000000000000	.....
00007ffe:6206b838	00007ffe6206c17e	~.b...
00007ffe:6206b840	0000000000000000	.....
00007ffe:6206b848	00007ffe6206c185	.b...
00007ffe:6206b850	00007ffe6206c1ad	.b...
00007ffe:6206b858	00007ffe6206c1cf	.b...
00007ffe:6206b860	00007ffe6206c1ee	.b...
00007ffe:6206b868	00007ffe6206c215	.b...
00007ffe:6206b870	00007ffe6206c256	.b...
00007ffe:6206b878	00007ffe6206c277	w.b...

Stack Debugger Error Console

paused

Рисунок 4 – выполнение вычислений



Рисунок 5 – выполнение вычислений

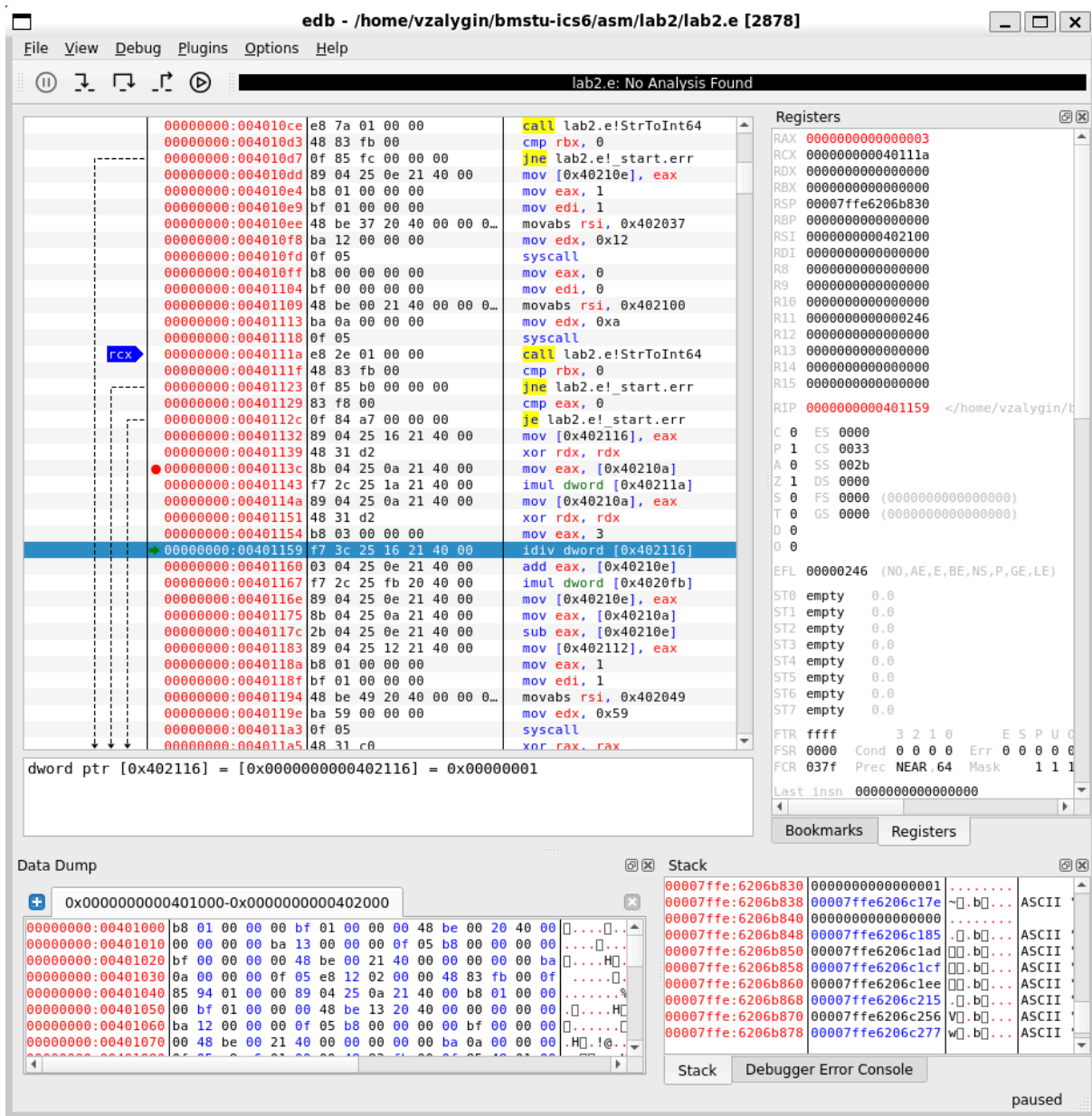


Рисунок 6 – выполнение вычислений



Рисунок 7 – выполнение вычислений



Рисунок 8 – выполнение вычислений



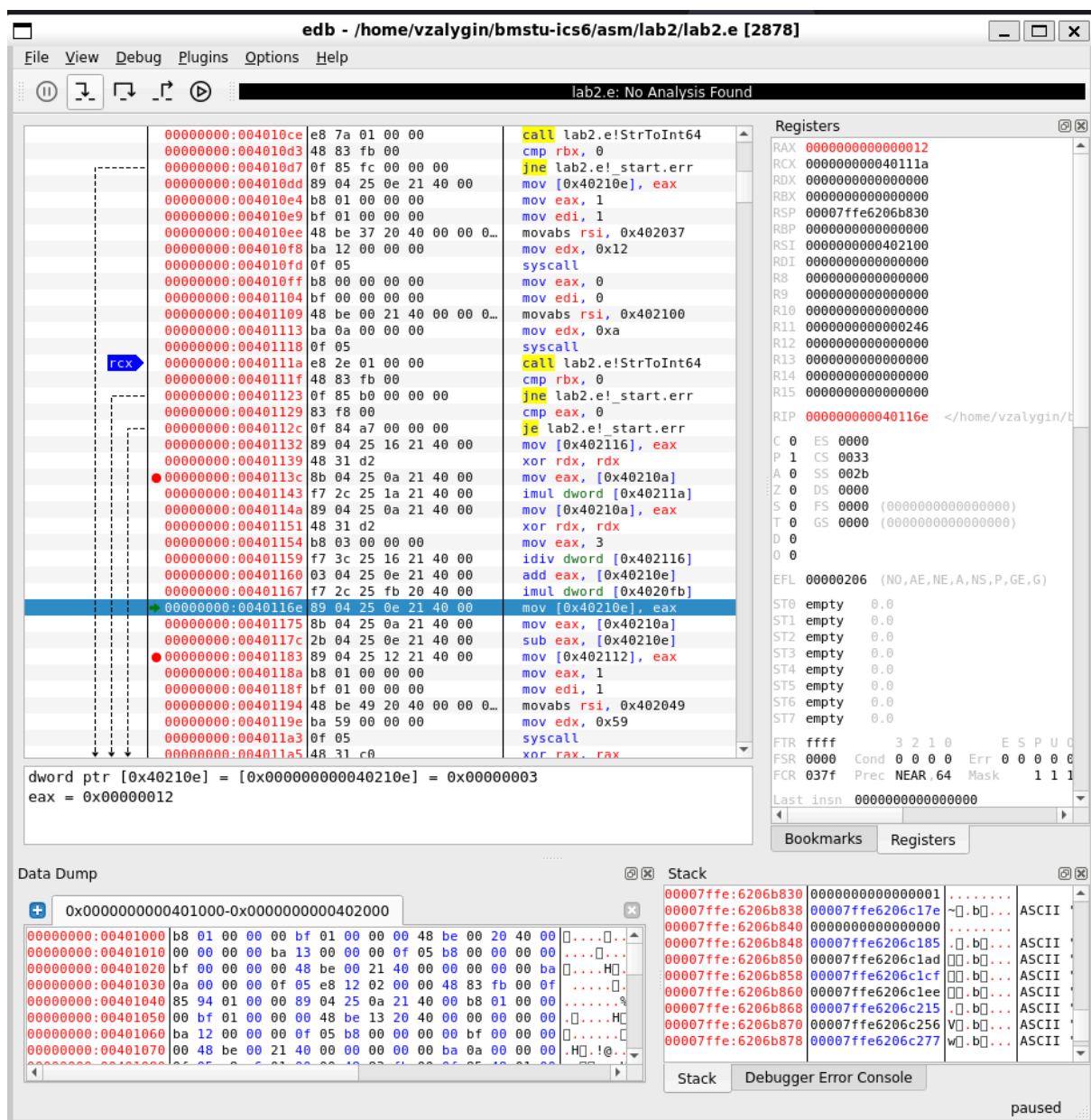


Рисунок 9 – выполнение вычислений

edb - /home/vzalygin/bmstu-ics6/asm/lab2/lab2.e [2878]

File View Debug Plugins Options Help

lab2.e: No Analysis Found

Address	Disassembly	Comment
00000000:004010ce	e8 7a 01 00 00	call lab2.e!StrToInt64
00000000:004010d3	48 83 fb 00	cmp rbx, 0
00000000:004010d7	0f 85 fc 00 00 00	jne lab2.e!_start.err
00000000:004010dd	89 04 25 0e 21 40 00	mov [0x40210e], eax
00000000:004010e4	b8 01 00 00 00	mov eax, 1
00000000:004010e9	bf 01 00 00 00	mov edi, 1
00000000:004010ee	48 be 37 20 40 00 00 0...	movabs rsi, 0x402037
00000000:004010f8	ba 12 00 00 00	mov edx, 0x12
00000000:004010fd	0f 05	syscall
00000000:004010ff	b8 00 00 00 00	mov eax, 0
00000000:00401104	bf 00 00 00 00	mov edi, 0
00000000:00401109	48 be 00 21 40 00 00 0...	movabs rsi, 0x402100
00000000:00401113	ba 0a 00 00 00	mov edx, 0xa
00000000:00401118	0f 05	syscall
00000000:0040111a	e8 2e 01 00 00	call lab2.e!StrToInt64
00000000:0040111f	48 83 fb 00	cmp rbx, 0
00000000:00401123	0f 85 b0 00 00 00	jne lab2.e!_start.err
00000000:00401129	83 f8 00	cmp eax, 0
00000000:0040112c	0f 84 a7 00 00 00	je lab2.e!_start.err
00000000:00401132	89 04 25 16 21 40 00	mov [0x402116], eax
00000000:00401139	48 31 d2	xor rdx, rdx
00000000:0040113c	8b 04 25 0a 21 40 00	mov eax, [0x40210a]
00000000:00401143	f7 2c 25 1a 21 40 00	imul dword [0x40211a]
00000000:0040114a	89 04 25 0a 21 40 00	mov [0x40210a], eax
00000000:00401151	48 31 d2	xor rdx, rdx
00000000:00401154	b8 03 00 00 00	mov eax, 3
00000000:00401159	f7 3c 25 16 21 40 00	idiv dword [0x402116]
00000000:00401160	03 04 25 0e 21 40 00	add eax, [0x40210e]
00000000:00401167	f7 2c 25 fb 20 40 00	imul dword [0x4020fb]
00000000:0040116e	89 04 25 0e 21 40 00	mov [0x40210e], eax
00000000:00401175	8b 04 25 0a 21 40 00	mov eax, [0x40210a]
00000000:0040117c	2b 04 25 0e 21 40 00	sub eax, [0x40210e]
00000000:00401183	89 04 25 12 21 40 00	mov [0x402112], eax
00000000:0040118a	b8 01 00 00 00	mov eax, 1
00000000:0040118f	bf 01 00 00 00	mov edi, 1
00000000:00401194	48 be 49 20 40 00 00 0...	movabs rsi, 0x402049
00000000:00401199	ba 59 00 00 00	mov edx, 0x59
00000000:004011a3	0f 05	syscall
00000000:004011a5	48 31 c0	xor rax, rax

dword ptr [0x40210a] = [0x000000000040210a] = 0x00000008  
eax = 0x00000012

Registers

Register	Value
RAX	0000000000000012
RCX	0000000000004011a
RDX	0000000000000000
RBX	0000000000000000
RSP	00007ffe6206b830
RBP	0000000000000000
RSI	000000000000402100
RDI	0000000000000000
R8	0000000000000000
R9	0000000000000000
R10	0000000000000000
R11	0000000000000246
R12	0000000000000000
R13	0000000000000000
R14	0000000000000000
R15	0000000000000000
RIP	000000000000401175 </home/vzalygin/t...
C 0	ES 0000
P 1	CS 0033
A 0	SS 002b
Z 0	DS 0000
S 0	FS 0000 (0000000000000000)
T 0	GS 0000 (0000000000000000)
D 0	
O 0	
EFL	00000206 (NO,AE,NE,A,NS,P,GE,G)
ST0	empty 0.0
ST1	empty 0.0
ST2	empty 0.0
ST3	empty 0.0
ST4	empty 0.0
ST5	empty 0.0
ST6	empty 0.0
ST7	empty 0.0
FTR	ffff 3 2 1 0 E S P U O
FSR	0000 Cond 0 0 0 0 Err 0 0 0 0
FCR	037f Prec NEAR,64 Mask 1 1 1
Last insn	0000000000000000

Bookmarks Registers

Data Dump

0x0000000000401000-0x0000000000402000

Address	Disassembly	Comment
00000000:00401000	b8 01 00 00 bf 01 00 00 48 be 00 20 40 00	
00000000:00401010	00 00 00 ba 13 00 00 0f 05 b8 00 00 00	
00000000:00401020	bf 00 00 00 48 be 00 21 40 00 00 00 ba	
00000000:00401030	0a 00 00 0f 05 e8 12 02 00 48 83 fb 00 0f	
00000000:00401040	85 94 01 00 00 89 04 25 0a 21 40 00 b8 01 00 00	
00000000:00401050	00 bf 01 00 00 00 48 be 13 20 40 00 00 00 00	
00000000:00401060	ba 12 00 00 0f 05 b8 00 00 00 bf 00 00 00	
00000000:00401070	00 48 be 00 21 40 00 00 00 ba 0a 00 00 00	

Stack

Address	Disassembly	Comment
00007ffe:6206b830	0000000000000001	
00007ffe:6206b838	00007ffe6206c17e	~.b...
00007ffe:6206b840	0000000000000000	
00007ffe:6206b848	00007ffe6206c185	.b...
00007ffe:6206b850	00007ffe6206c1ad	~.b...
00007ffe:6206b858	00007ffe6206c1cf	~.b...
00007ffe:6206b860	00007ffe6206c1ee	~.b...
00007ffe:6206b868	00007ffe6206c215	~.b...
00007ffe:6206b870	00007ffe6206c256	V.b...
00007ffe:6206b878	00007ffe6206c277	w.b...

Stack Debugger Error Console

paused

Рисунок 10 – выполнение вычислений



Рисунок 11 – выполнение вычислений



edb - /home/vzalygin/bmstu-ics6/asm/lab2/lab2.e [2878]

File View Debug Plugins Options Help

lab2.e: No Analysis Found

Registers

RAX 00000000ffffff6  
RCX 000000000040111a  
RDX 0000000000000000  
RBX 0000000000000000  
RSP 00007ffe6206b830  
RBP 0000000000000000  
RSI 0000000000402100  
RDI 0000000000000000  
R8 0000000000000000  
R9 0000000000000000  
R10 0000000000000000  
R11 00000000000000246  
R12 0000000000000000  
R13 0000000000000000  
R14 0000000000000000  
R15 0000000000000000  
RIP 0000000000401183 </home/vzalygin/t

48 be 00 21 40 00 00 00 00

call lab2.e!StrToInt64  
cmp rbx, 0  
jne lab2.e!\_start.err  
mov [0x40210e], eax  
mov eax, 1  
mov edi, 1  
movabs rsi, 0x402037  
mov edx, 0x12  
syscall  
mov eax, 0  
mov edi, 0  
movabs rsi, 0x402100  
mov edx, 0xa

dword ptr [0x402112] = [0x0000000000402112] = 0x00000000  
eax = 0xffffffff6

Data Dump

0x0000000000401000-0x0000000000402000

Stack

00007ffe:6206b830 0000000000000001 ~.b... ASCII  
00007ffe:6206b838 00007ffe6206c17e ~.b... ASCII  
00007ffe:6206b840 0000000000000000 ~.b... ASCII  
00007ffe:6206b848 00007ffe6206c185 ~.b... ASCII  
00007ffe:6206b850 00007ffe6206c1ad ~.b... ASCII  
00007ffe:6206b858 00007ffe6206c1cf ~.b... ASCII  
00007ffe:6206b860 00007ffe6206c1ee ~.b... ASCII  
00007ffe:6206b868 00007ffe6206c215 ~.b... ASCII  
00007ffe:6206b870 00007ffe6206c256 ~.b... ASCII  
00007ffe:6206b878 00007ffe6206c277 ~.b... ASCII

Stack Debugger Error Console

paused

Рисунок 12 – выполнение вычислений



Рисунок 13 – выполнение вычислений

## Расшифровка кодов мнемоники mov

1)

4 8 8 9 c 2  
 1001000 100010 01 11 000 010  
 префикс 1 100010dw mod reg reg  
 mov rdx, rax

2)

8 9 0 4 2 5 16 21 40 00

10001001 00 000100 001 00 101 00010110 00100001 0100000 000000000

100010dw mod reg mem ss mod 16 21 40 00

mov [0x402116], eax

3)

b a 1 2 00 00 00

10111010 00010010 00000000 00000000 00000000

101110dw 12 00 00 00

mov edx, 0x12

### Тестирование

Результаты тестирования приведены в таблице 1.

Таблица 1 – результаты тестирования

№	Поток ввода	Поток вывода	Вердикт
1	2 4 3 1	-10	Верно
2	1 1 1 1	-11	Верно
3	1000 2000 3000 4000	24920	Верно

### Вывод

Изучены форматы машинных команд, команд целочисленной арифметики ассемблера, а также выполнено программирование целочисленных вычислений.

### **Ответы на контрольные вопросы**

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

Машинная команда представляет собой код, определяющий операцию вычислительной машины и данные, участвующие в операции. Префиксы разделяют на:

- префикс повторения – используется только для строковых команд;
- префикс размера адреса (67h) – применяется для изменения размера смещения: 16 бит при 32-х разрядной адресации;
- префикс размера операнда (66h) – указывается, если вместо 32-х разрядного регистра для хранения операнда используется 16-ти разрядный;
- префикс замены сегмента – используется при адресации данных любым сегментом кроме DS.

d – направление обработки, например, пересылки данных: 1 – в регистр, 0 – из регистра;

w – размер операнда: 1 – операнды - двойные слова, 0 – операнды байты;

mod – режим:

- 1) 00 - Disp=0 – смещение в команде 0 байт;
- 2) 01 - Disp=1 – смещение в команде 1 байт;
- 3) 10 - Disp=2 – смещение в команде 2 байта;
- 4) 11 - операнды-регистры. Регистры кодируются в зависимости от размера операнда:

Регистры кодируются в зависимости от размера операнда:

- 1) для w = 1:

– 000 EAX;

- 001 ECX;
- 010 EDX;
- 011 EBX;
- 100 ESP;
- 101 EBP;
- 110 ESI;
- 111 EDI.

2) для  $w = 0$ :

- 000 AL;
- 001 CL;
- 010 DL;
- 011 BL;
- 100 AH;
- 101 CH;
- 110 DH;
- 111 BH.

2) Назовите мнемоники основных команд целочисленной арифметики.

Какие форматы для них можно использовать?

Мнемоники: add, sub, div/idiv, mul/imul. Арифметические операции, такие как сложение, вычитание, деление и умножение можно выполнять над однобайтовыми, двухбайтовыми и четырёхбайтовыми целыми числами.

3) Сформулируйте основные правила построения линейной программы вычисления заданного выражения.

- Объявить инициализированные переменные.
- Объявить неинициализированные переменные.
- Написать команды для вычисления выражений арифметических действий и команды перемещения, объявленных ранее переменных как в регистры, так и из них.

- Написать процедур ввода и вывода, если этого требует условие поставленного выражения.

Арифметические операции, такие как сложение, вычитание, деление и умножение можно выполнять над однобайтовыми, двухбайтовыми и четырёхбайтовыми целыми числа.

- 4) Почему ввод-вывод на языке ассемблера не программируют с использованием соответствующих машинных команд? Какая библиотека используется для организации ввода вывода в данной лабораторной?

Так как для выполнения арифметических операций числа должны быть представлены в памяти в одном из внутренних форматов, в которых знак, если он предусмотрен, кодируется первым битом, а само число записано в двоичной системе счисления.

Используется библиотека, состоящая из функций, IntToStr, StrToInt.

- 5) Расскажите, какие операции используют при организации ввода-вывода.

Перевод строки в число и перевод числа в строку. Вызовы системных функция ввода-вывода.