



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, представленное на рисунке 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

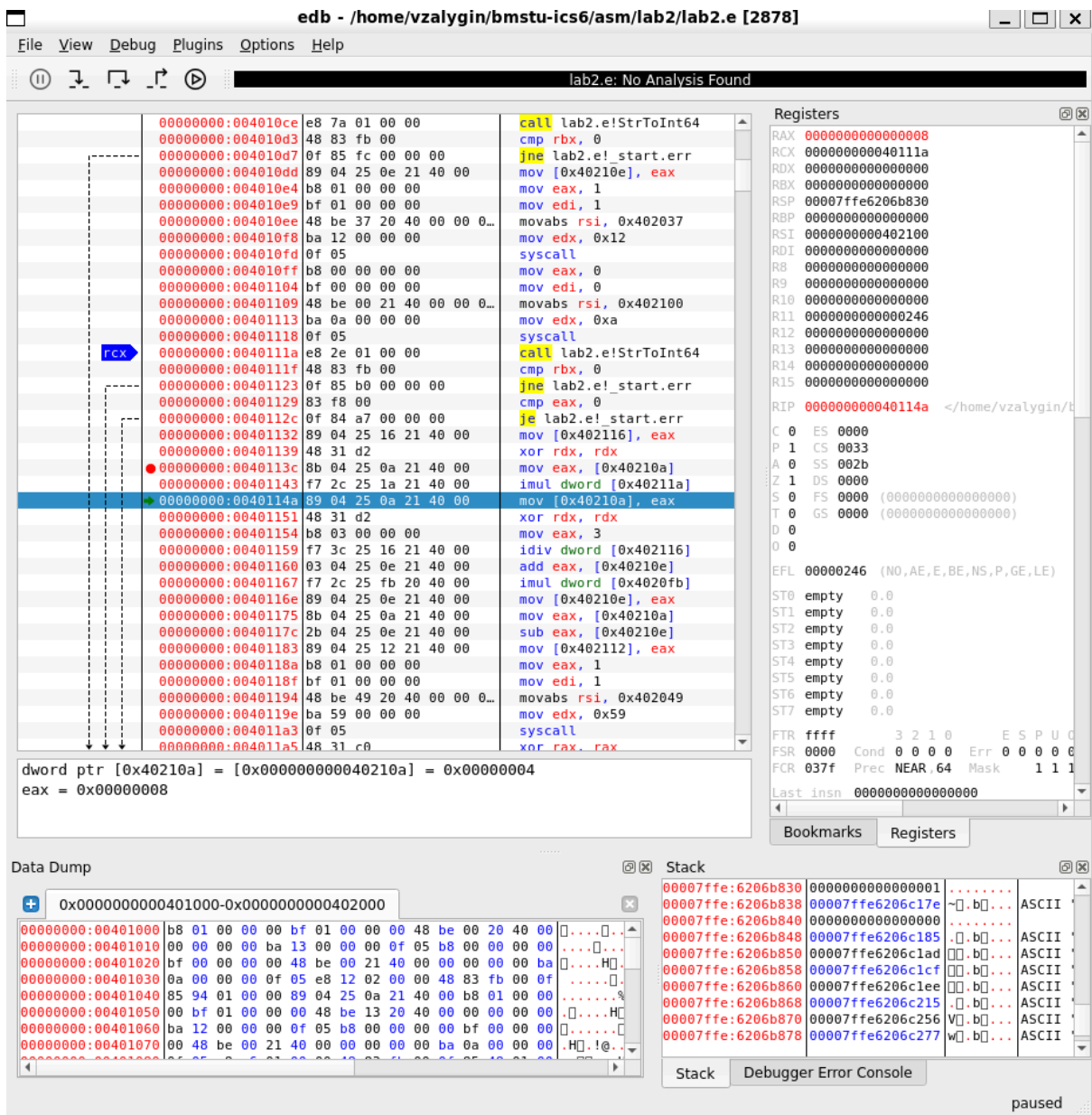


Рисунок 4 – Выполнение вычислений



Рисунок 5 – Выполнение вычислений



Рисунок 6 – Выполнение вычислений

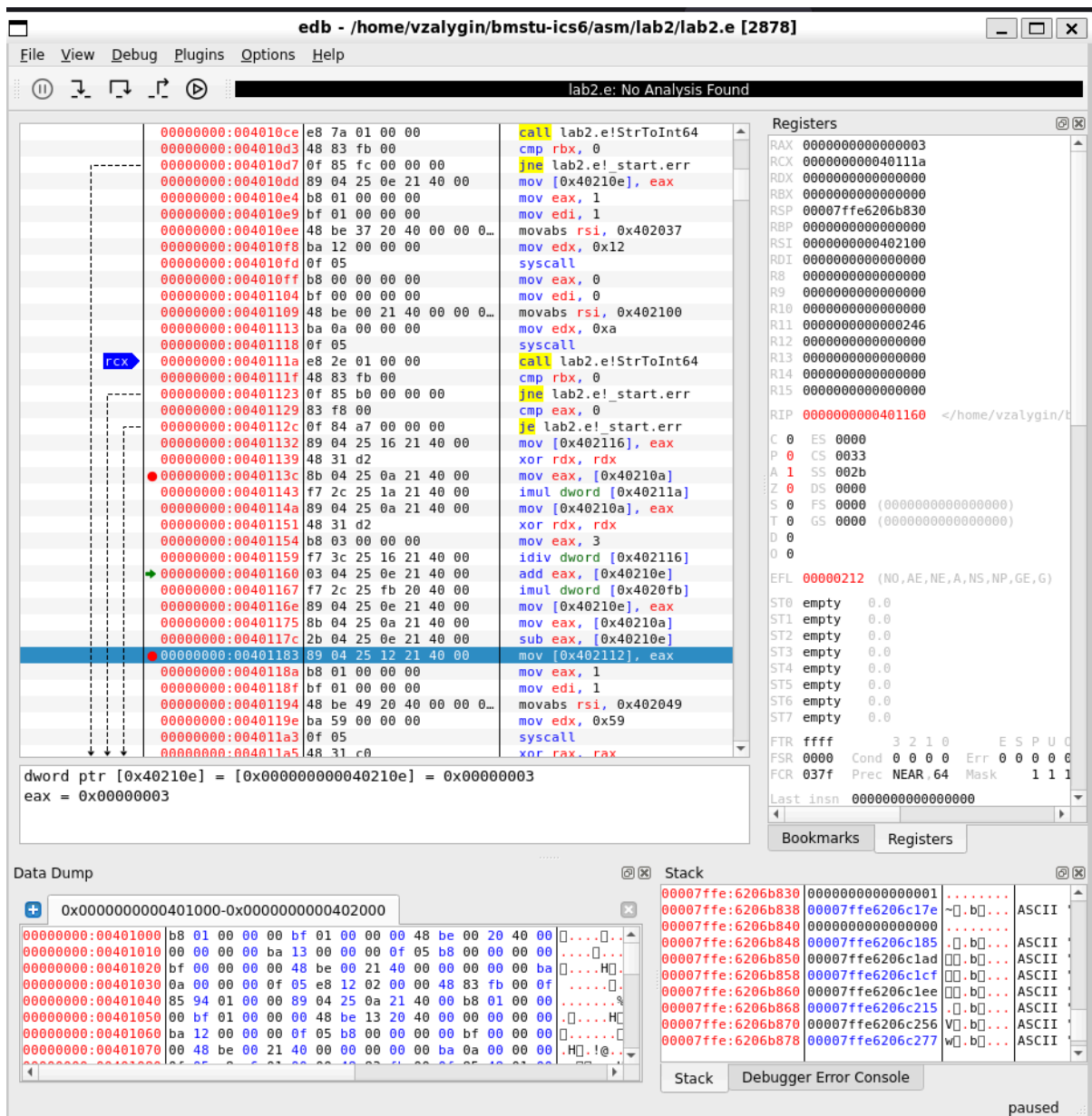


Рисунок 7 – Выполнение вычислений



Рисунок 8 – Выполнение вычислений

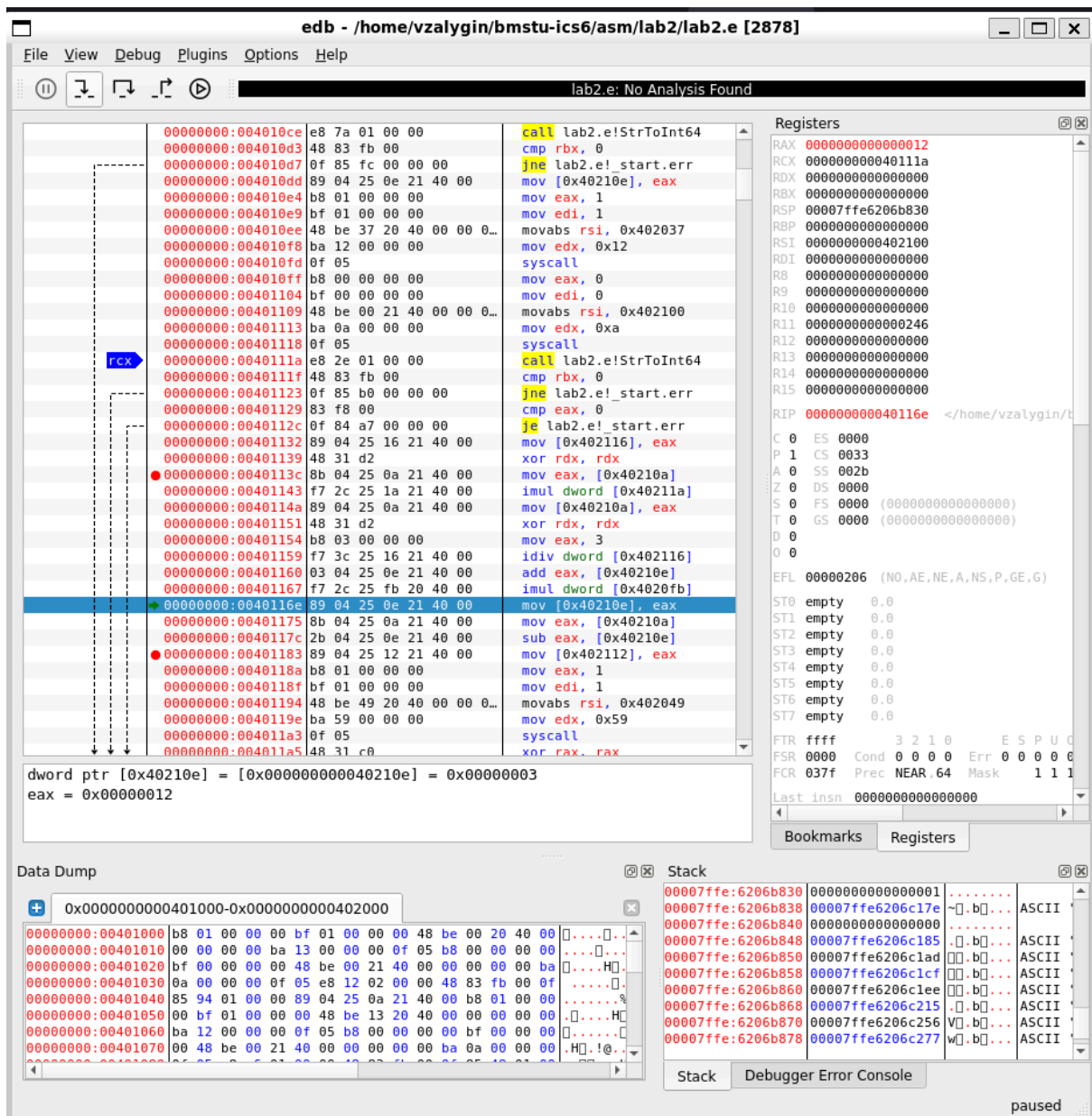


Рисунок 9 – Выполнение вычислений



Рисунок 10 – Выполнение вычислений

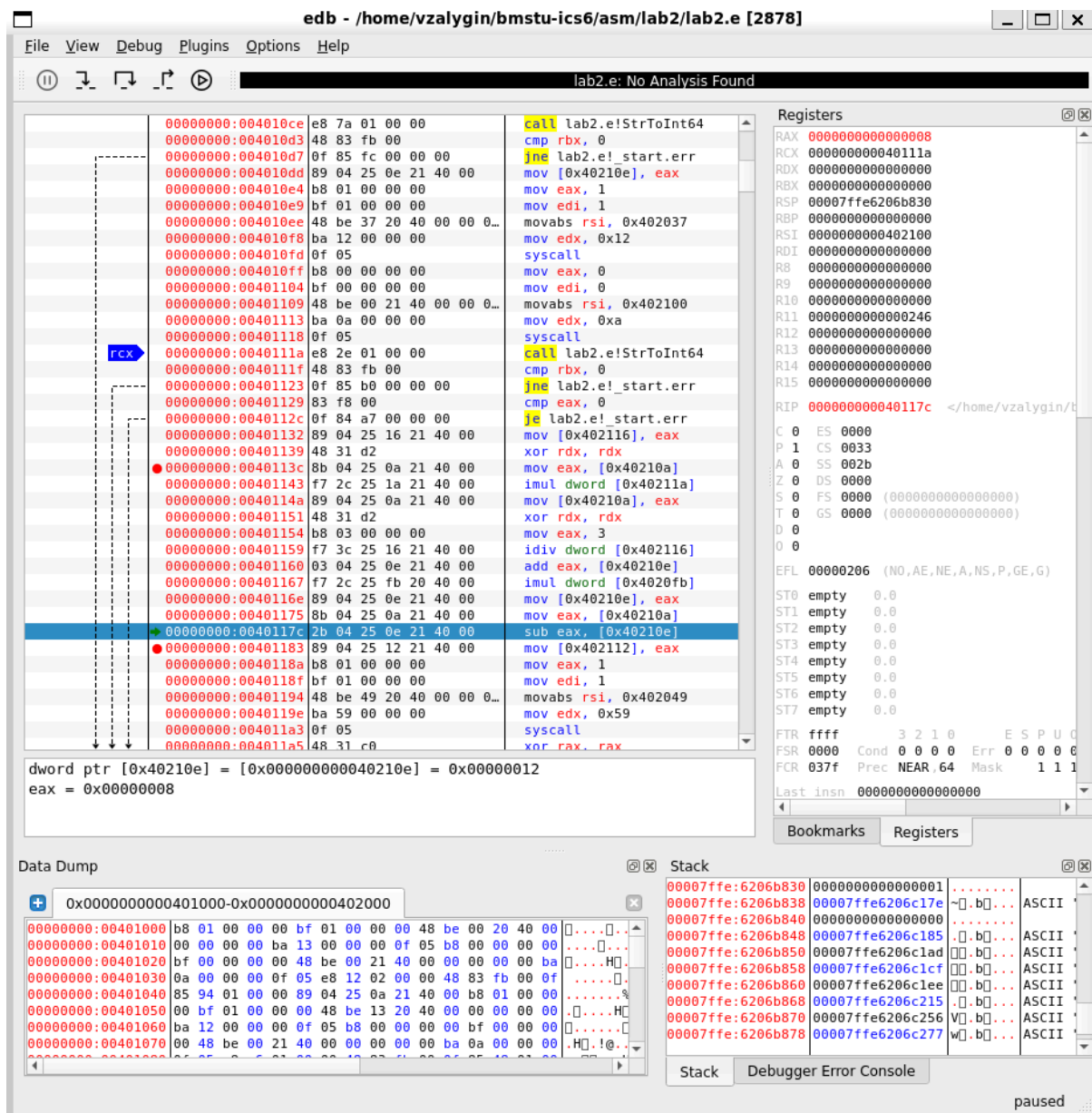


Рисунок 11 – Выполнение вычислений

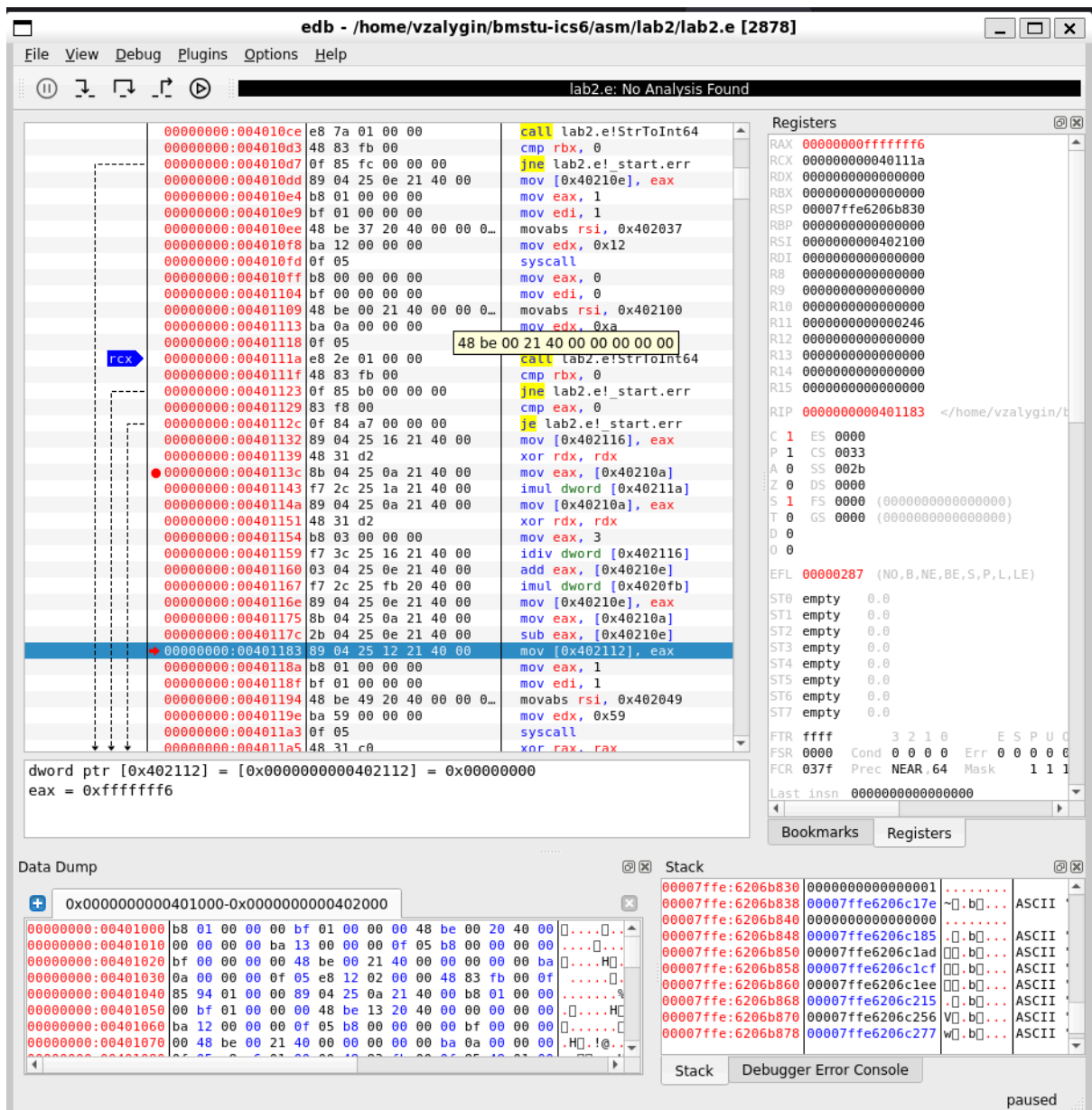


Рисунок 12 – Выполнение вычислений



Рисунок 13 – Выполнение вычислений

Расшифровка кодов мнемоники mov

1) Первый код

4 8 8 9 с 2

1001000 100010 01 11 000 010

префикс 1 100010dw mod reg reg

mov rdx, rax

2) Второй код

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) Третий код

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/ldiv, 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 числами.

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 функций ввода-вывода.