**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО**

**ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №1

Дисциплина: Архитектура вычислительных систем

по теме Разработка программ на ассемблере.

Работа с отладчиком OllyDbg и пакетом masm32

Выполнил: ст. группы ВТ-31  
Новожен Н.В

Проверил: Осипов.О.В

**Белгород 2019**

**Цель работы:** получить навыки создания и отладки простейших программ на

ассемблере с использованием пакета masm32 и отладчика OllyDbg.

Задания для выполнения к работе

1. Ознакомиться со средой OllyDbg и компилятором masm32.
2. Создать и скомпилировать программу в соответствии с вариантом задания.
3. Отладить программу.
4. С помощью OllyDbg определить местонахождение переменных в сегменте

данных, а также их размер.

1. Выполнить пошаговую трассировку программы. Определить какие регистры
2. изменяют свои значения в процессе выполнения команд.

**Выполнения работы**

1. Создать файл lab1.asm со следующим содержимым:

.386

.MODEL FLAT, STDCALL

OPTION CASEMAP: NONE

include E:\masm32\include\windows.inc

include E:\masm32\include\kernel32.inc

include E:\masm32\include\user32.inc

includelib E:\masm32\lib\user32.lib

includelib E:\masm32\lib\kernel32.lib

.DATA

strs DB 10, 13, "Summ:", 0

sum DD ?

mas DW 4 DUP(5)

longFloat DQ 154.5

.CODE

START:

MOV EAX, sum

MOV EBX, 4

MOV ESI, 0

ADD AX, mas[ESI]

ADD ESI, 2

MUL EBX

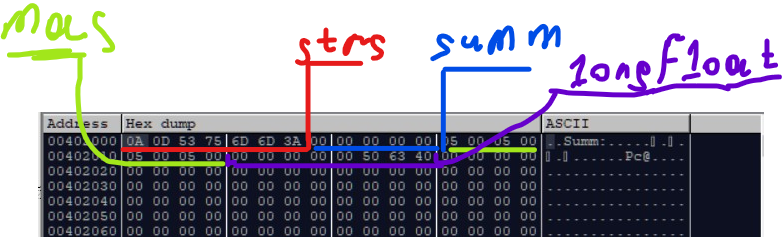
MOV sum, EAX

push NULL

call ExitProcess

END START

1. Скомпилировать программу и получить исполняемый файл lab1.exe.
2. Открыть файл lab1.exe в OllyDbg.
3. Сегмент данных содержит одну строку, один массива и три переменные:

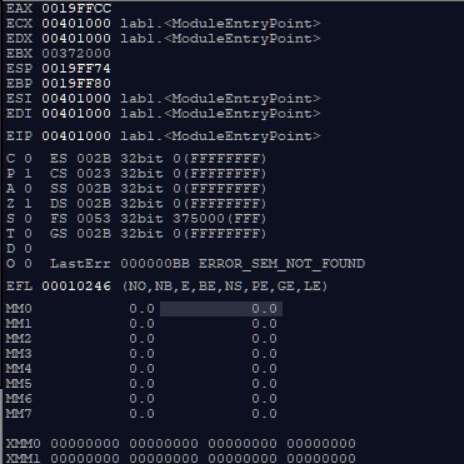


154.5=10011010.1000; 50=00110010 ;63=0111111 ;40=00101000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название переменной | Начальный адрес | Конечный адрес | Размер данных, байт | Тип |
| **Strs** | 00402002 | 00402006 | 5 | строка |
| **–** | 00402000 | 00402001 | 2 | символы возврата каретки (10, 13) |
| **-** | 00402007 | 00402007 | 1 | символы окончания строки (0) |
| **Sum** | 00402008 | 0040200B | 4 | 4-байтовое целое |
| **Mas** | 0040200C | 00402013 | 8 | массив из четырёх двухбайтовых целых чисел |
| **longFloat** | 00402014 | 0040201B | 8 | 8-байтовое целое |
| Общий размер сегмента данных: | | | **28** |  |

1. Пошаговое выполнение программы

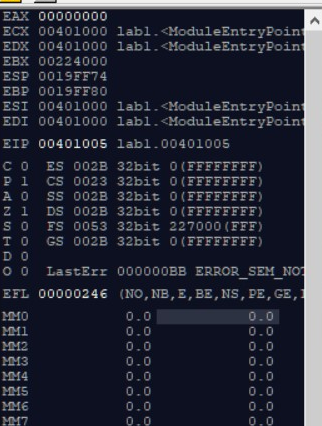
Исходное состояние регистров:



* 1. MOV EAX, sum

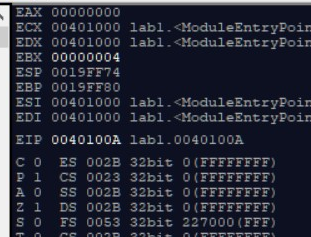
Код команды – **A108204000**. Команда пересылает в регистр **EAX** 4 байта из ячейки памяти с адресом 00402008-0040200B

(переменная **sum**). Увеличивает **EIP** на 5 (размер кода команды).



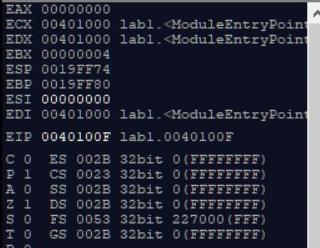
5.2 MOV EBX, 4

Код команды – **BB04000000**. Команда пересылает в регистр **EBX** константу 4 . Увеличивает **EIP** на 5 (размер кода команды).

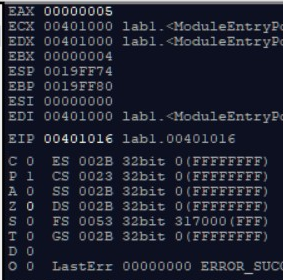


5.3 MOV ESI, 0

Код команды – **BE00000000**. Команда пересылает в регистр **ESI** константу 0 . Увеличивает **EIP** на 5 (размер кода команды).

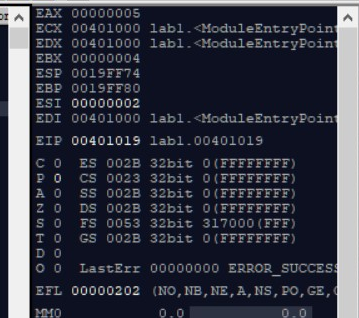


5.4 ADD AX, mas[ESI]

Код команды – **66:0386 OC2040000**. Команда прибавляет к регистру **AX** число из памяти по адресу 00402009h. По данному адресу хранится первый элемент массива **mas** – число 5. Результат – **AX** равен 5 . **EIP** увеличивается на 7. Изменяются флаг Z. 

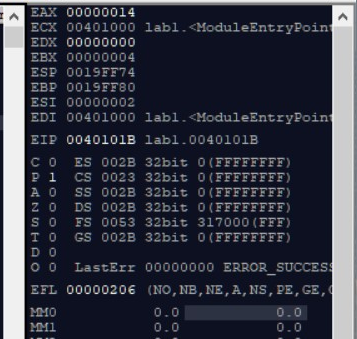
5.5 ADD ESI, 2

Код команды – **83C6 02**. Команда прибавляет к регистру **ESI** число 2. Результат – **ESI** равен 2 . **EIP** увеличивается на 3. Изменяются флаг P.



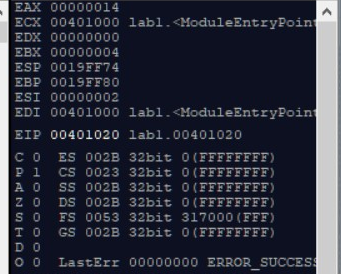
5.6 MUL EBX

Код команды – **F7E3**. Команда умножает к регистру **EAX** на EBX. Результат – **EAXAX** равен 20(14h) . **EIP** увеличивается на 2. Изменяются флаг P.



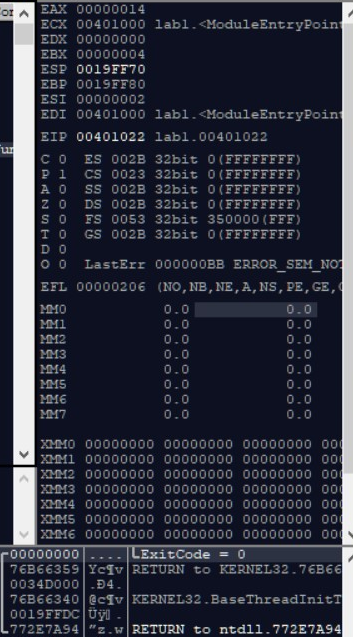
5.7 MOV sum, EAX

Код команды – A30820400. Команда пересылает в sum данные из EAX . Увеличивает **EIP** на 9 (размер кода команды).



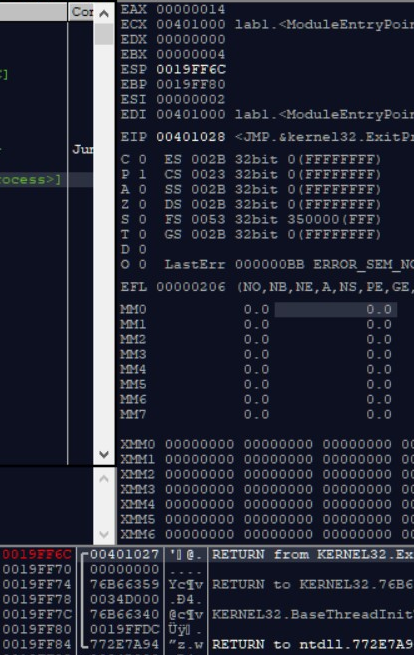
5.8 push NULL

Код команды – 6A00. Команда передает в стек 0. Увеличивает **EIP** на 2 (размер кода команды).



5.9 call ExitProcess

Код команды – E8 01000000. Команда завершает процесс. Увеличивает **EIP** на 6 (размер кода команды).



Вывод: В ходе лабараторной работы мы научились разрабатывать программу на ассемблере работать с отладчиком OllyDbg и пакетом masm32. Познакомились с регистрами процессора,способом вызова функций и как расположены данные в сегменте данных.