**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ **«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. В. Г. ШУХОВА»**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Дисциплина: Архитектура вычислительных систем**

**Тема: Разработка программ на ассемблере. Работа с отладчиком OllyDbg и пакетом masm32**

Выполнил: ст. группы ВТ-31

Подкопаев Антон Валерьевич

Проверил: Осипов Олег Васильевич

**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.

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

**Вариант 13**

.DATA

scanf\_s DB "%d", 0

das DW 500

mas DD 8 DUP(1)

ten DT 30000000

.CODE

START:

XOR EAX, EAX

MOV AX, das

MUL 5

MOV EBX mas[0]

MOV das, AX

END START

**Ход выполнения работы**

.386

.MODEL FLAT, STDCALL

OPTION CASEMAP: NONE

include c:\masm32\include\windows.inc

include c:\masm32\include\kernel32.inc

include c:\masm32\include\user32.inc

includelib c:\masm32\lib\user32.lib

includelib c:\masm32\lib\kernel32.lib

.DATA

string DB "%d", 0

num DW 500

mas DD 8 DUP(1)

ten DT 30000000

.CODE

START:

XOR EAX, EAX

MOV AX, num ; das

MOV BX, 5

MUL BX

MOV EBX, mas[0]

MOV num, AX

PUSH NULL

CALL ExitProcess

END START

Создадим файл lab1.asm , скомпилируем программу и получим исполняемый файл lab1.exe. Откроем файл lab1.exe в OllyDbg.

Сегмент данных содержит строку, массив и две переменные:

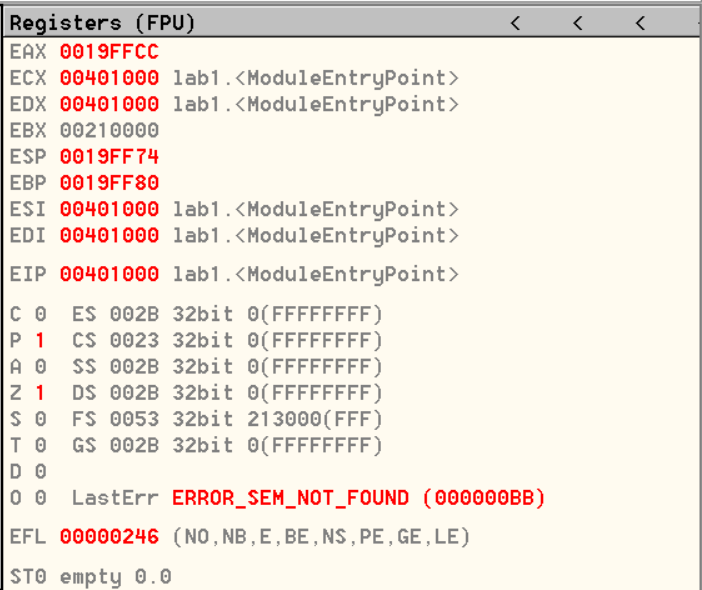
A screenshot of a cell phone

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название переменной | Начальный адрес | Конечный адрес | Размер данных, байт | Тип |
| **string** | 00403000 | 00403001 | 2 | строка |
| **–** | 00403002 | 00403002 | 1 | символ окончания строки (0) |
| **num** | 00403013 | 00403014 | 2 | двухбайтовое целое |
| **mas** | 00403015 | 00403024 | 32 | массив из восьми 4-байтовых целых чисел |
| **ten** | 00403025 | 0040302E | 10 | 10-байтовое целое |
| Общий размер сегмента данных: | | | **47** |  |

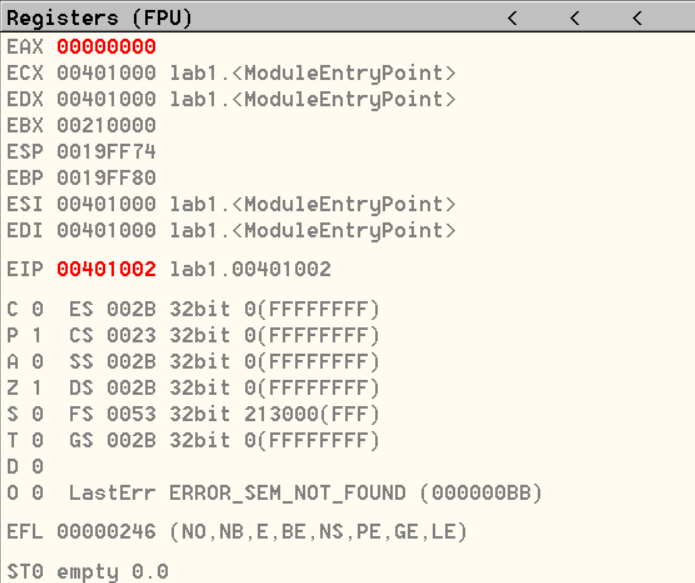
**Пошаговое выполнение программы**

Исходное состояние регистров:



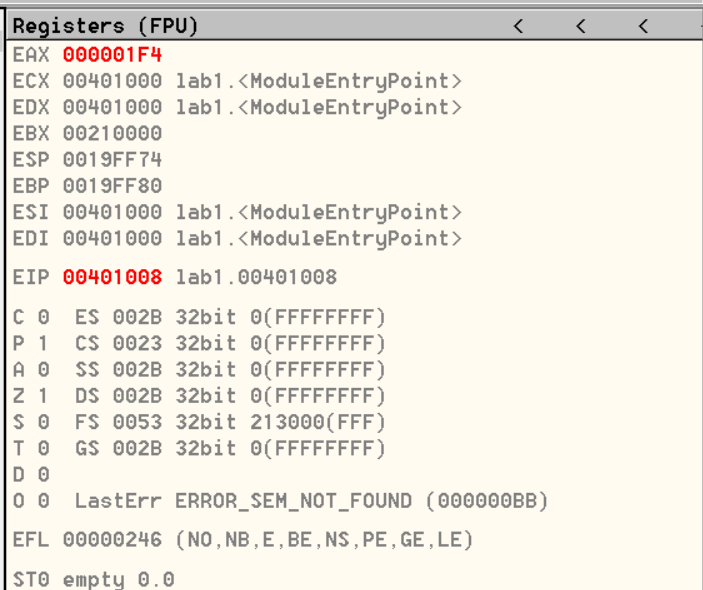
**XOR EAX, EAX** Код команды - **33С0**

Команда выполняет побитовое исключающее **ИЛИ** над регистром **EAX**, тем самым обнуляя его. Содержимое регистра **EIP** увеличивается на 2.



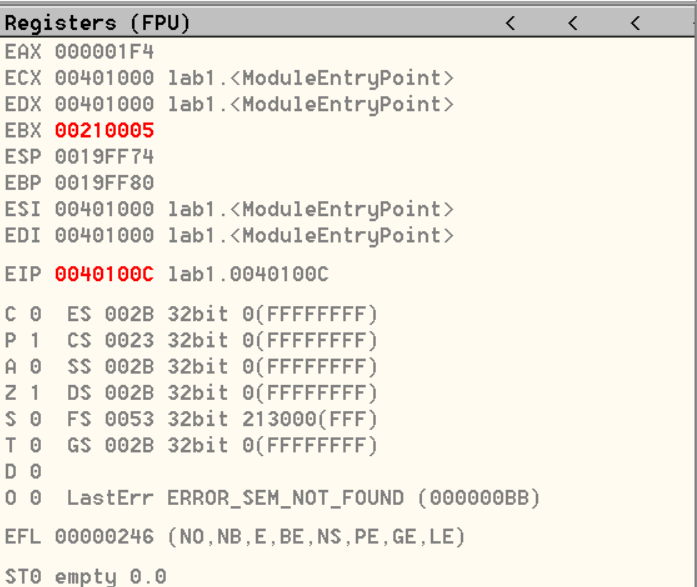
**MOV AX, num** Код команды - **66:A1 03304000**

Команда пересылает в регистр **AX** (часть **EAX**) байт из ячейки памяти с адресом 00403013h (переменная num = 500). Размер пересылаемых данных – два байта (слово). Увеличивает **EIP** на 5 (размер кода команды).



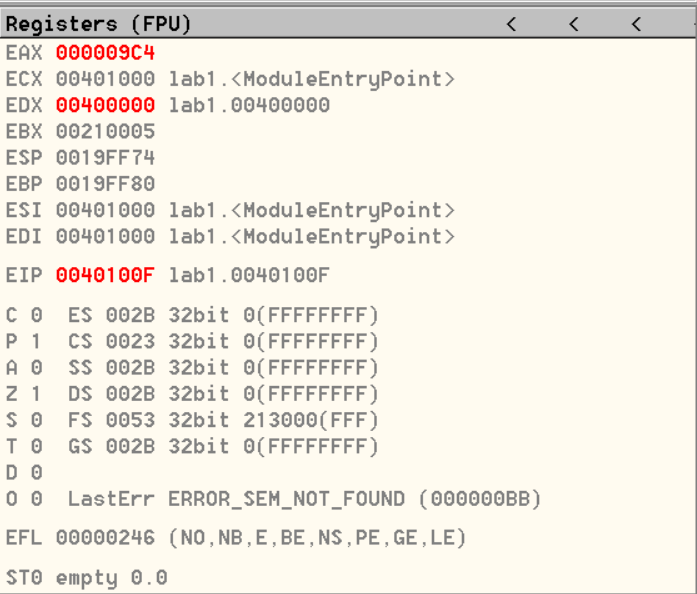
**MOV BX, 5** Код команды - **66:BB 0500**

Команда пересылает в регистр **BX** (часть **EBX**) число 5. Увеличивает **EIP** на 4 (размер кода команды).



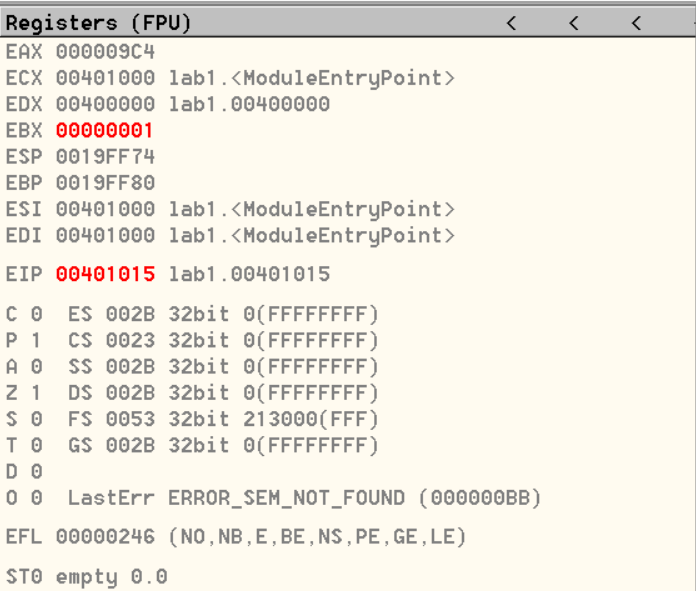
**MUL BX** Код команды **66:F7E3**

Команда перемножит содержимое регистров **AX** и **BX** (500 \* 5), а затем запишет результат в **AX**. Увеличивает **EIP** на 3 (размер кода команды).



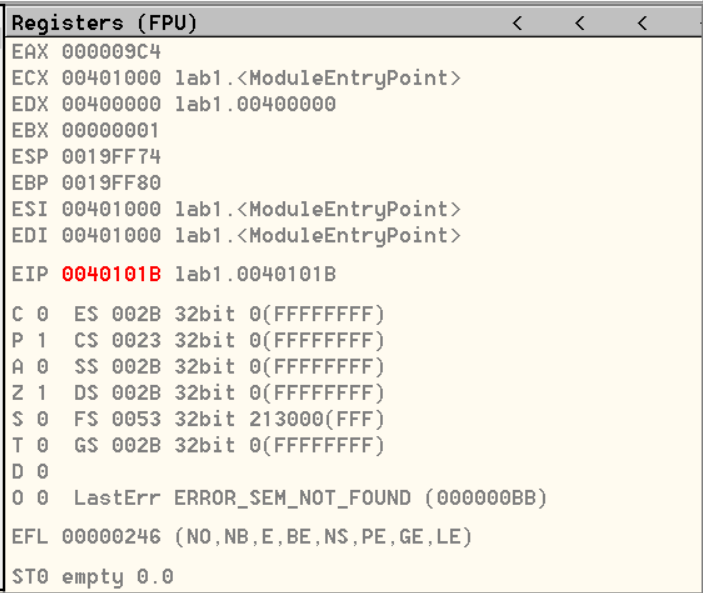
**MOV EBX, mas[0]** Код команды **8B1D 05304000**

Команда пересылает в регистр **EBX** два байта из ячейки памяти с адресом 00403015h (первый элемент массива mas[0] = 1). Размер пересылаемых данных – два байта (слово). Увеличивает **EIP** на 6 (размер кода команды).



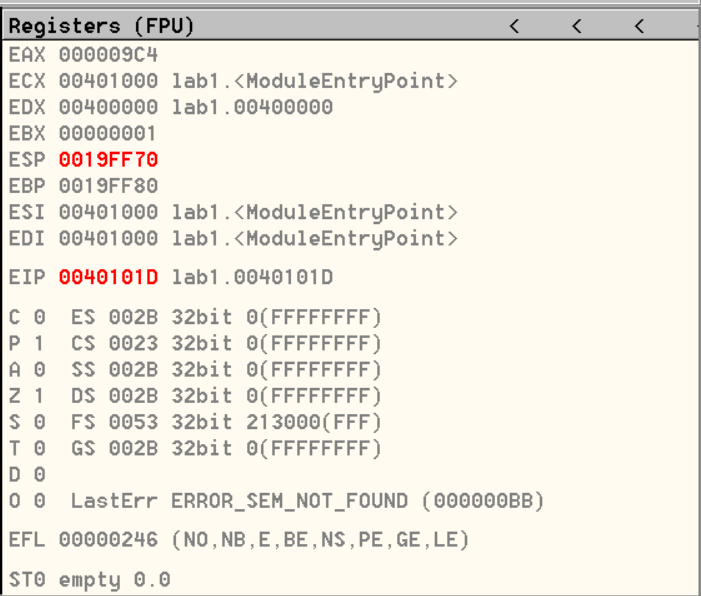
**MOV num, AX** Код команды **66:A3 03304000**

Команда записывает в сохраненную под **num** область памяти с адресом 00403013h содержимое регистра **AX**. Размер пересылаемых данных – 1 байт (слово). Увеличивает **EIP** на 6 (размер кода команды).



**PUSH NULL** Код комманды **6A 00**

Команда передает в стек 0, увеличивает **EIP** на 2 (размер кода команды).



**CALL ExitProcess** Код команды **E8 00000000**

Команда завершает процесс работы программы. Увеличивает **EIP** на 5 (размер кода команды).

