МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

Кафедра «Информационные системы»

**ОТЧЕТ**

о выполнении лабораторной работы №6

по дисциплине “Технические средства информационных систем”

Выполнил: ст. гр. ИС/б-20-2-о

Белик Г. М.

Проверил:

Минкин С. И.

Севастополь

2022

**ЛАБОРАТОРНАЯ РАБОТА №6**

**«ИССЛЕДОВАНИЕ СПОСОБОВ АНАЛИЗА ФУНКЦИОНИРОВАНИЯ 32-РАЗРЯДНЫХ ПРОЦЕССОРОВ С ПОМОЩЬЮ ОТЛАДЧИКА OLLYDBG»**

**Цель работы**

Исследование архитектуры 32–разрядных процессоров и возможностей отладчика OllyDbg при разработке и отладке ассемблерных программ, приобретение практических навыков по анализу и отладке программ на языке ассемблера.

**Постановка задачи**

1. Повторить теоретический материал, касающийся структуры 32–разрядных микропроцессоров, включая суперскалярные, программно-доступных регистров и системы команд языка ассемблера (выполняется при домашней подготовке).

2. Создать приложение типа Win32 с графическим интерфейсом, вычисляющее сумму и разность двух чисел: первое число – номер в группе, второе – число, противоположное номеру первой буквы фамилии в алфавите (отрицательное число в дополнительном коде). Противоположным к некоторому х называется число, равное х по абсолютной величине, но обратное по знаку.

3. Загрузить разработанное приложение в отладчик.

4. Исследовать работу программы в отладчике OllyDbg.

5. Проанализировать, какие изменения происходят в различных регистрах процессора:

− изменяйте значения регистров EAX и EBX таким образом, чтобы произошли изменения состояний флагов O, P, Z, S, C;

− поместите значение в стек с помощью инструкции PUSH; − извлеките значение из стека с помощью инструкции POP;

− изучите принцип работы инструкции MOV c 8–ми, 16–ти и 32–х разрядными регистрами;

− изучите работу математических инструкций: INC, DEC, ADD, SUB, MUL, IMUL, DIV, NEG;

− измените значение регистра ESP;

− измените значение регистра EIP;

− найдите инструкцию PUSH 0 и замените ее на NOP.

6. Результаты исследований отразите в отчете к данной работе.

**Ход работы**

Для начала выполнения работы требуется создать приложение типа Win32 с консольным интерфейсом, вычисляющее сумму и разность двух чисел: первое число – номер в группе, второе – число, противоположное номеру первой буквы фамилии в алфавите (отрицательное число в дополнительном коде).

Программа была написана на языке C++, текст приведен ниже:

#include <iostream>

#include <stdint.h>

int main()

{

int32\_t a = 22;

int32\_t b = -20;

int32\_t task1 = a+b;

std::cout << "Sum task:"<<task1;

int32\_t task2 = a-b;

std::cout <<"\n Division task:"<<task2;

}

Разработанное приложение было загружено в отладчик OllyDbg. Затем, было исследовано приложение и проанализированы изменения в различных регистрах процессора. Результаты представлены на рисунках 1-4:

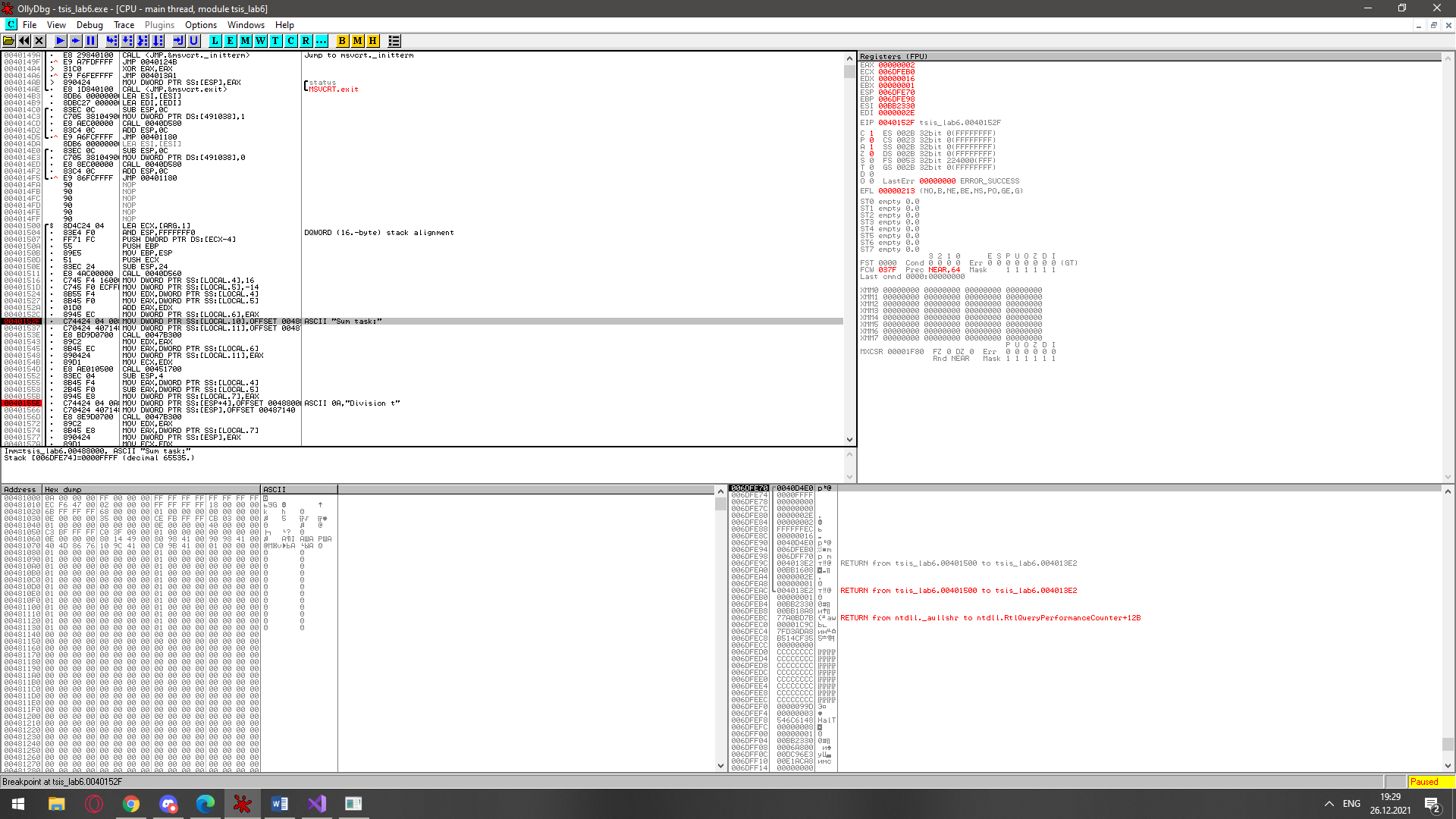


Рисунок 1 – Выполнение первой арифметической операции (результат 2)

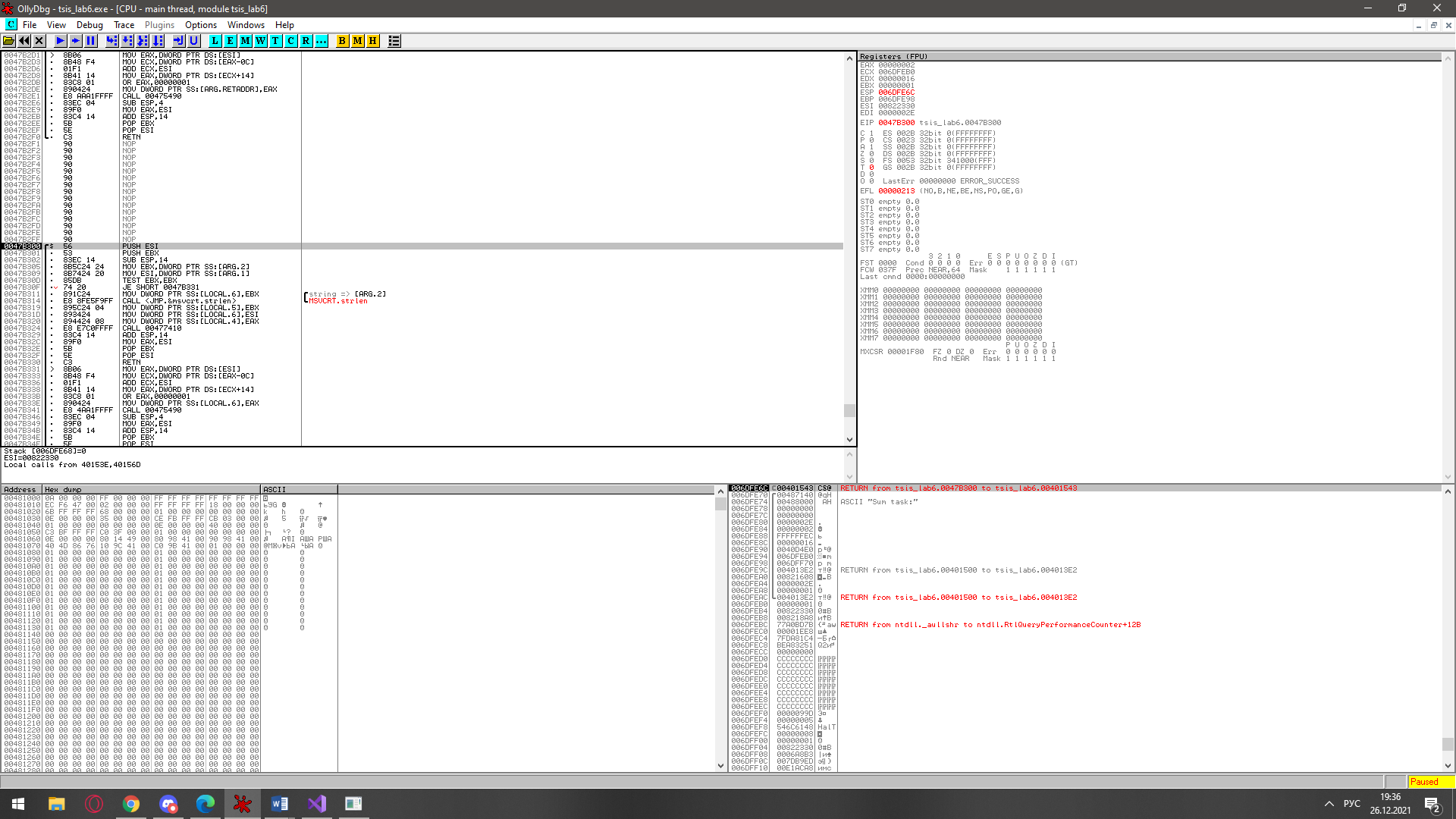


Рисунок 2 – Вход в подпрограмму потокового вывода (переход в подпрограмму call)

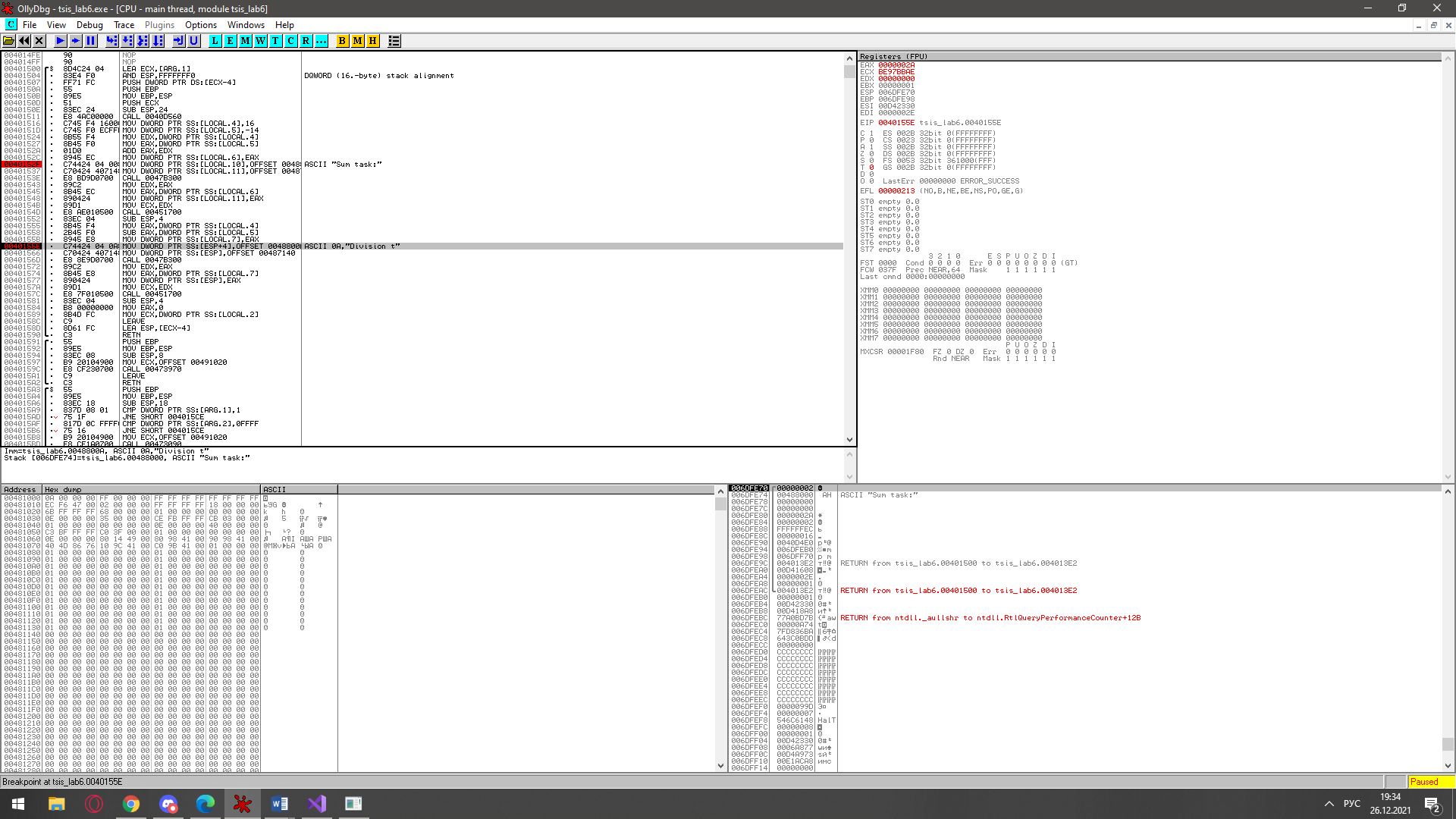


Рисунок 3 – Выполнение второй арифметической операции (результат 42)

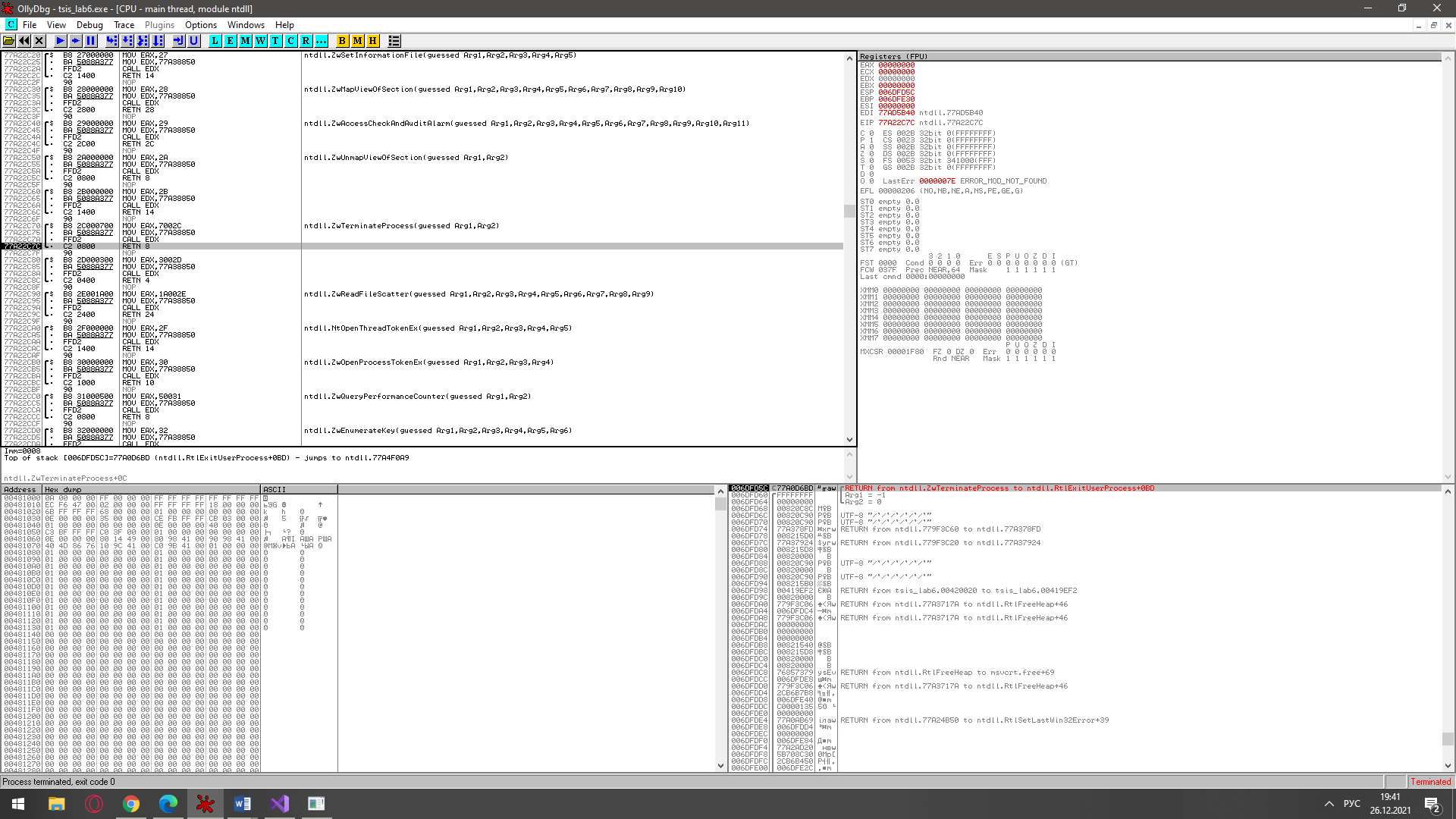


Рисунок 4 – Завершение работы

Работа программы была исследована в отладчике.

Для изменения флага O, значение регистра EAX было изменено на 7FFFFFFF, после чего была выполнена команда ADD EAX, 1, в результате чего флаг переполнения O и флаг знака S (т.к. произошло переполнение и число стало отрицательным) стали равны 1 (Рисунок 5).

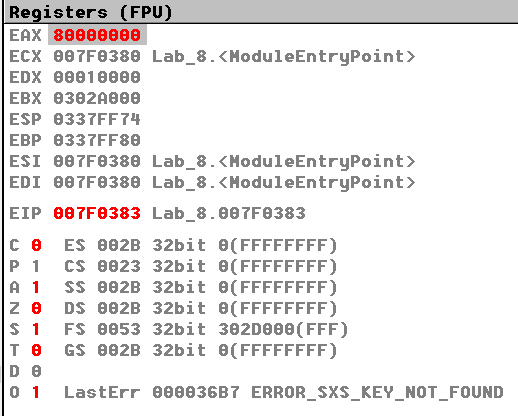


Рисунок 5 – Флаг переполнения, четности и флаг знака равны 1

После выполнения предыдущей операции, флаг четности P равен 1, т.к. число единиц в числе нечетно. При помощи команды ADD EAX, 1, увеличим регистр EAX на 1, тем самым изменив флаг P (Рисунок 6).

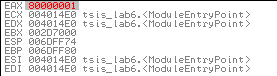


Рисунок 6 – Флаг знака равен 0

Регистр EAX был установлен на FFFFFFFF (-1 в десятичной записи) (Рисунок 7)

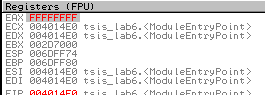


Рисунок 7 – Регистр EAX равен -1

После этого была выполнена команда ADD EAX, 1, в результате чего был получен ноль и произошёл перенос и флаги нуля Z и переноса C стали равны 1 (Рисунок 8).

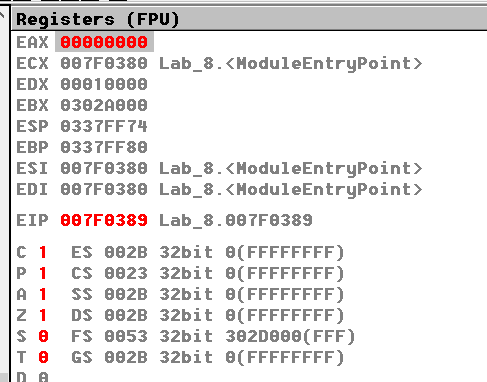


Рисунок 8 – Флаги Z и C равны 1

При помощи команды PUSH EAX, в стек было занесено значение регистра EAX, предварительно выставленное и равное 111 в шестнадцатеричной системе (Рисунок 9).



Рисунок 9 – Значение 111 в фрейме стека

При помощи команды POP EBX, значение 101 из стека было выгружено в регистр EBX (Рисунок 10).

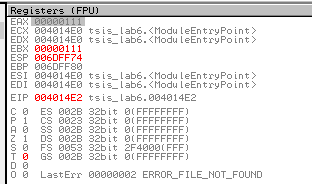


Рисунок 10 – Значение 101 из стека в регистре EBX

Были изучены математические операции INC, DEC, увеличивающие и уменьшающие регистры на единицу соответственно, операции ADD, SUB и MUL, складывающие, вычитающие и умножающие значения регистров.

Была использована команда IMUL EBX, которая умножает два числа и может дать результат до 64 бит длинной. Число 0FFFFFFF было записано в EAX и число 000FFFFF – в EBX, в результате выполнения команды умножения результат был записан в два регистра – EDX и EAX (Рисунок 11).

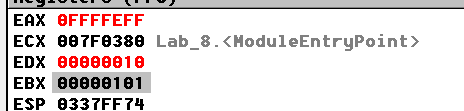


Рисунок 11 – Результат IMUL

При помощи функции Set origin here, было изменено значение регистра EIP (Рисунок 12).



Рисунок 12 – Изменение регистра EIP

Регистр ESP был вручную изменён на адрес стека, содержащий нули, при этом изменение отображается в отладчике (Рисунок 13).

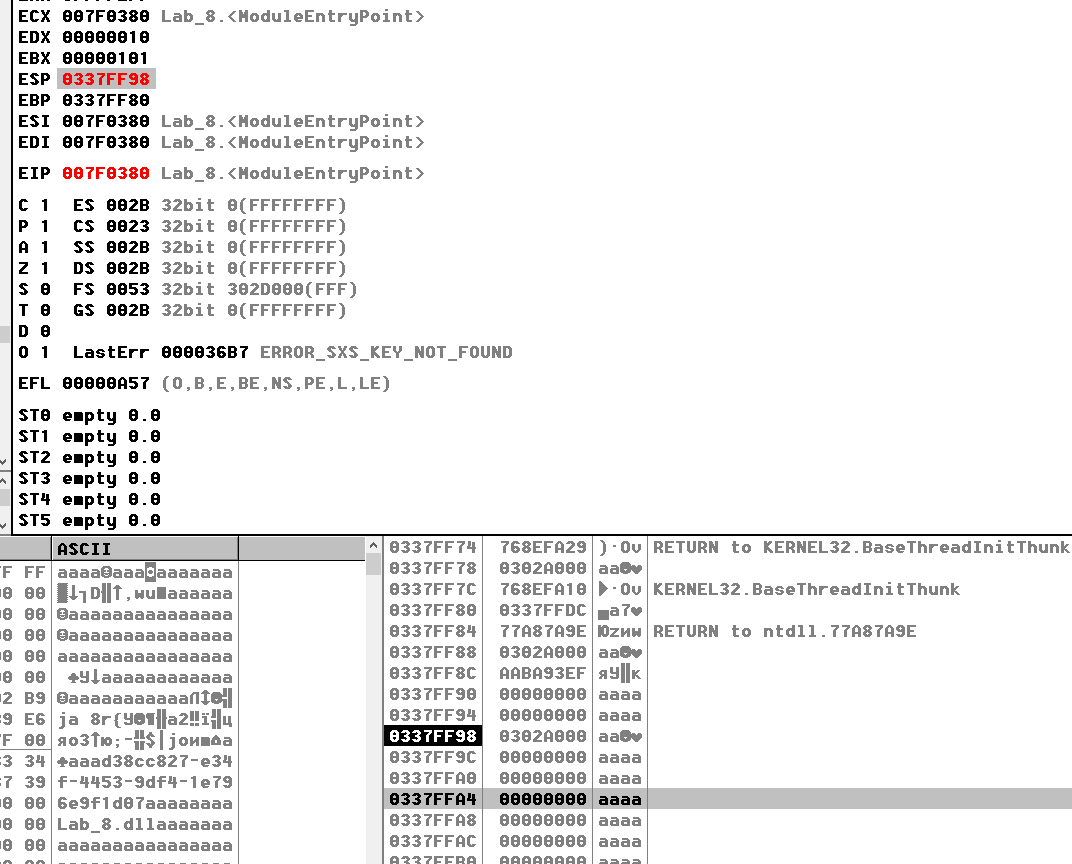


Рисунок 13 – Изменение регистра ESP

Была найдена команда PUSH 0 и заменена NOP (Рисунок 14). В результате место PUSH 0 заняли две команды NOP.

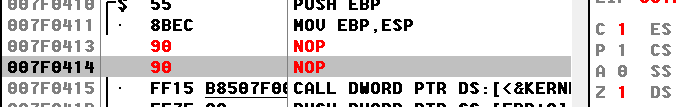
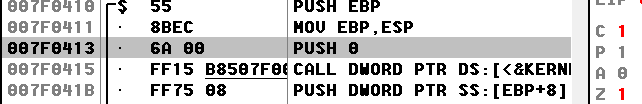


Рисунок 14 – Замена PUSH 0 на NOP

**Вывод**

В ходе работы были исследованы архитектуры 32-разрядных процессоров и возможности отладчика OllyDbg при разработке и отладке приложений Win32, приобретены практические навыки по анализу и отладке программ на языке ассемблера.