**ЛАБОРАТОРНА РОБОТА № 3**

з дисципліни

«Архітектура комп`ютерів»

Шейко Р. О.

Групи КІ-21-2

**Завдання на лабораторну роботу**

1. Згідно номеру свого варіанта (табл. 4.7) скласти програму організації циклів за допомогою команд умовного переходу і декременту. Порівняти з алгоритмом виконання конструкції while.

2. Організувати цикли за допомогою команди loop. Початкове значення регістру ЕСХ/СХ розраховувати з варіанту програмно.

3. Організувати цикли по типу алгоритму виконання конструкції do за допомогою команд асемблеру.

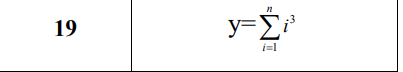


Рис 1. Варіант завдання

2. Код до програми першого завдання знаходиться у додатку А, другого завдання знаходиться у додатку Б, а третє у В.

3. Почнемо тестування. Почнемо з першої програми:

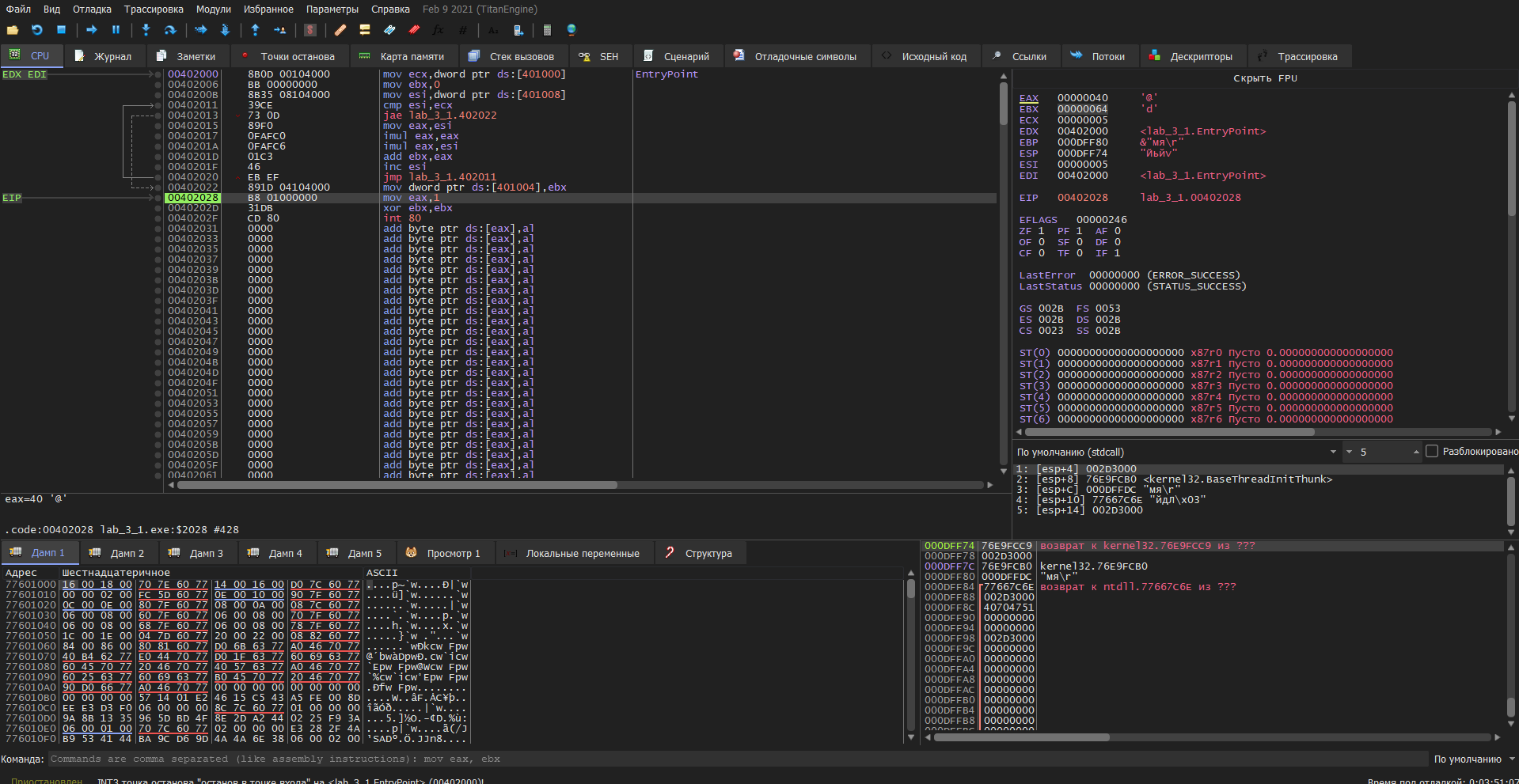


Рис 2. Перше завдання після завершення коду

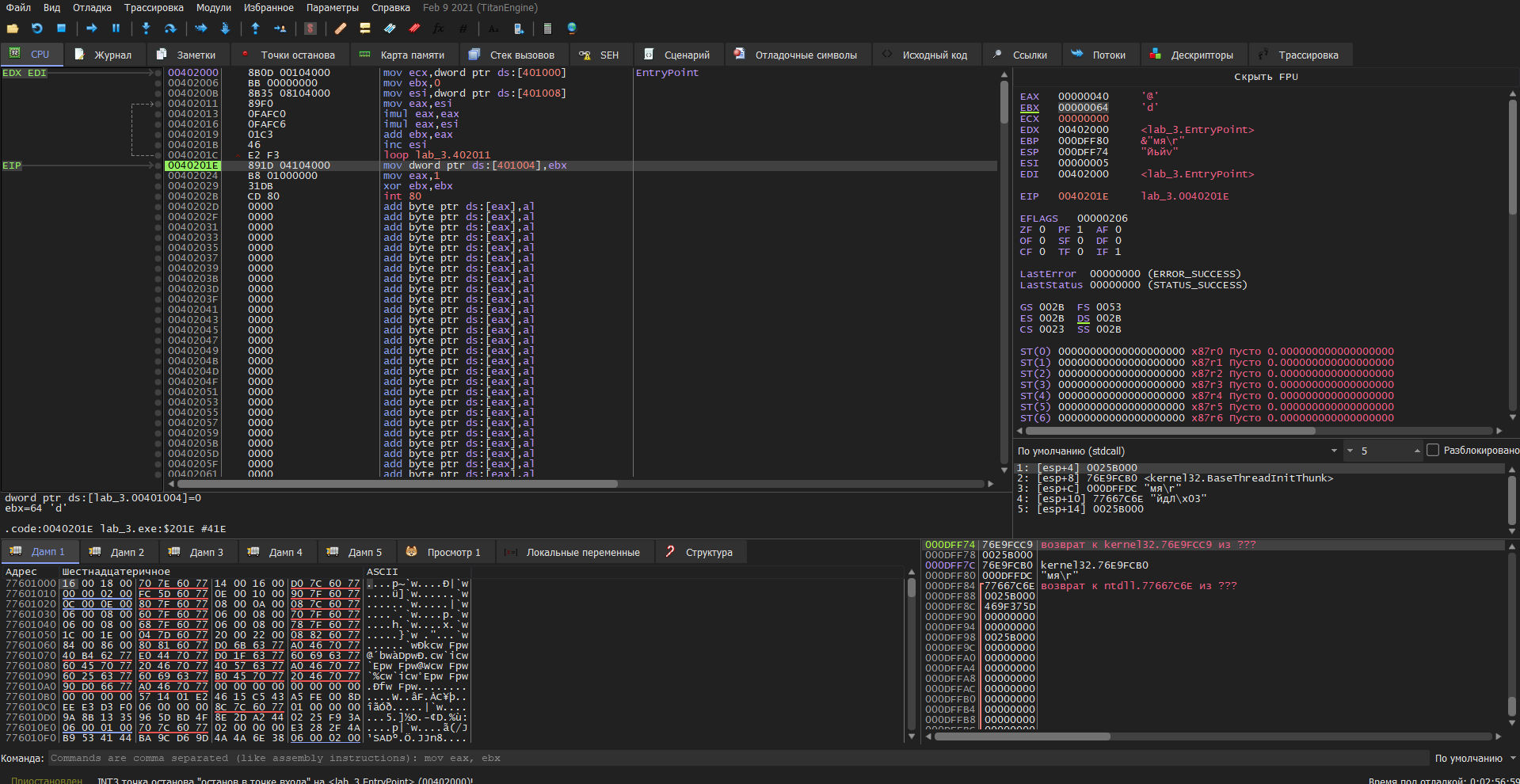


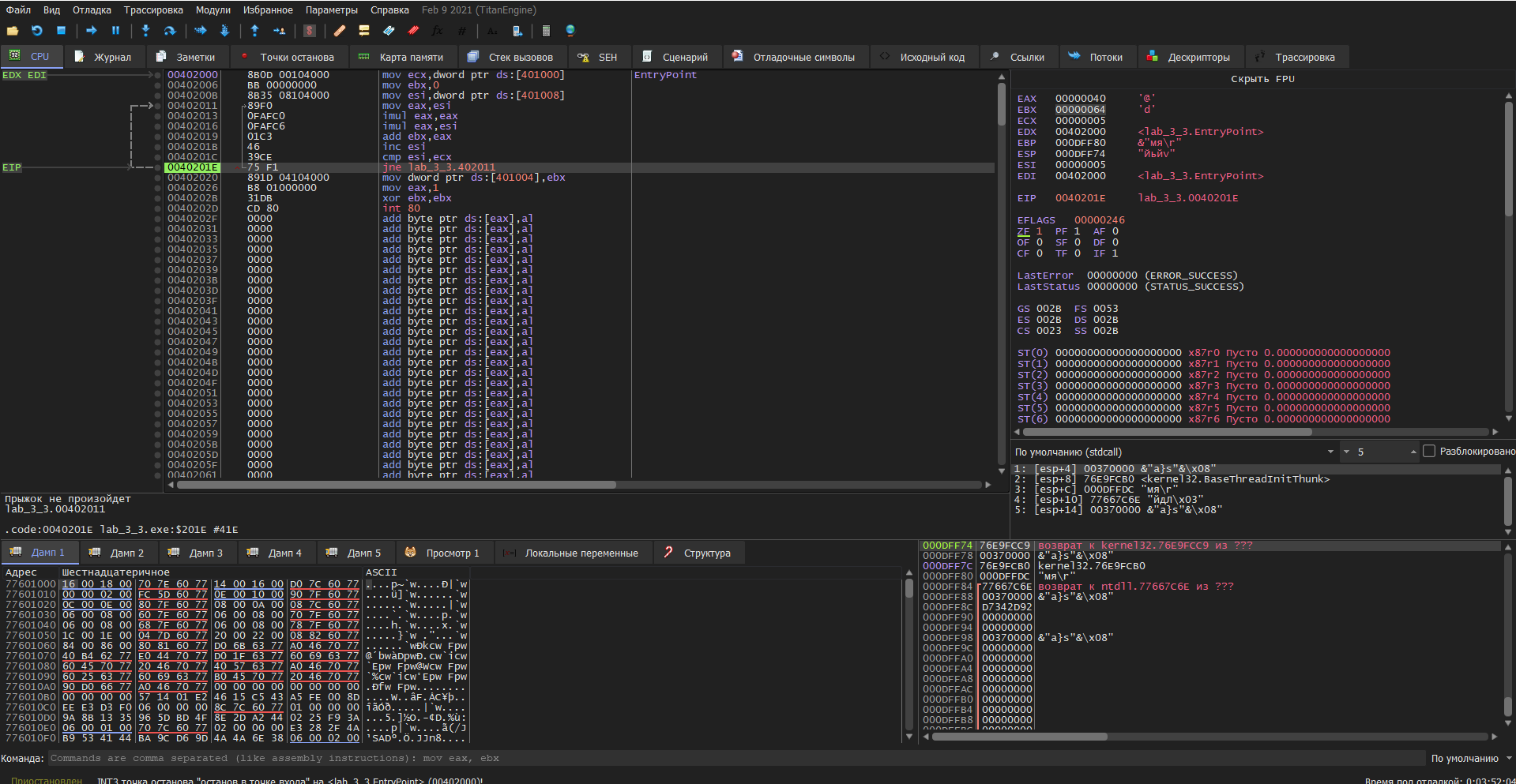
Рис 3. Друге завдання після завершення коду  


Рис 3. Третє завдання після завершення коду

Додаток А

*Лістинг завдання 1*

format PE console

entry start

section '.data' data readable writeable

n dd 5

sum dd 0

i dd 1

section '.code' code readable executable

start:

mov ecx, [n]

mov ebx, 0

mov esi, [i]

while\_start:

cmp esi, ecx

jae while\_end

mov eax, esi

imul eax, eax ; Обчислюємо куб числа

imul eax, esi

add ebx, eax

inc esi

jmp while\_start

while\_end:

mov [sum], ebx

mov eax, 1

xor ebx, ebx

int 0x80

Додаток Б

*Лістинг завдання 2*

*format PE console*

*entry start*

*section '.data' data readable writeable*

*n dd 17 ; Задаем значение n (количество чисел)*

*sum dd 0 ; Инициализируем переменную для хранения суммы*

*i dd 1 ; Инициализируем переменную для хранения текущего числа i*

*section '.code' code readable executable*

*start:*

*mov ecx, [n] ; Загружаем n в регистр ecx*

*mov ebx, 0 ; Обнуляем регистр ebx для хранения суммы*

*mov esi, [i] ; Загружаем начальное значение i в регистр esi*

*; Цикл для вычисления суммы кубов натуральных чисел*

*sum\_loop:*

*; Вычисление куба текущего числа и добавление к сумме*

*mov eax, esi*

*imul eax, eax ; Квадрат текущего числа*

*imul eax, esi ; Куб текущего числа*

*add ebx, eax*

*inc esi ; Увеличение i*

*loop sum\_loop*

*; Результат находится в регистре ebx (сумма кубов)*

*mov [sum], ebx*

*; Теперь в sum*

*; Завершение программы*

*mov eax, 1*

*xor ebx, ebx*

*int 0x80*

*;section '.idata' import data readable*

*;library kernel, 'kernel32.dll',\*

*; msvcrt, 'msvcrt.dll'*

*;import kernel,\*

*; ExitProcess, 'ExitProcess'*

*;import msvcrt,\*

*; printf, 'printf'*

Додаток В

*Лістинг завдання 3*

*format PE console*

*entry start*

*section '.data' data readable writeable*

*n dd 5*

*sum dd 0*

*i dd 1*

*section '.code' code readable executable*

*start:*

*mov ecx, [n]*

*mov ebx, 0*

*mov esi, [i]*

*sum\_loop:*

*mov eax, esi*

*imul eax, eax*

*imul eax, esi*

*add ebx, eax*

*inc esi*

*cmp esi, ecx*

*jne sum\_loop*

*mov [sum], ebx*

*mov eax, 1*

*xor ebx, ebx*

*int 0x80*