***Упражнение 504.*** На лабораторном занятии про MessageBox была решена задача по выводу его на экран. В исходном тексте есть команда:

**mov message+6,'h'**

Используя любые доступные Вам средства, указать:

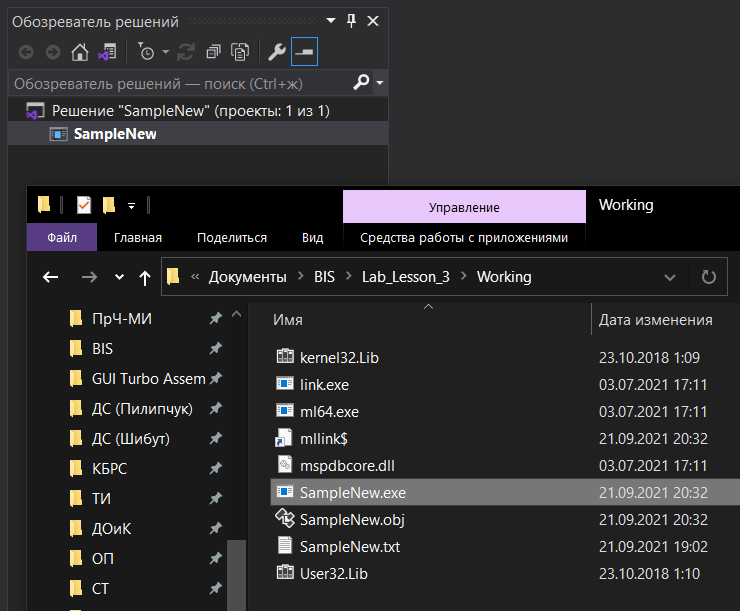
1. КАКОЙ АДРЕС имеет эта команда,
2. нарисовать шестнадцатеричное представление этой команды в памяти ЭВМ,
3. указать содержимое регистра EIP, когда исходный текст будет оттранслирован с получением объектного модуля, который в свою очередь будет далее пролинчеван с получением на внешнем устройстве исполнимого модуля, который в свою очередь будучи загружен операционной системой в оперативную память превратится в программу, которой будет передано управление, и когда в конце концов дойдёт дело до исполнения этой команды.
4. А также указать в шестнадцатеричном виде АДРЕС ОБЛАСТИ ПАМЯТИ, поименованной как message.

CDBYMZ: Под доступными средствами понимаются разного рода отладчики, трансляторы, получающие разного рода листинги.

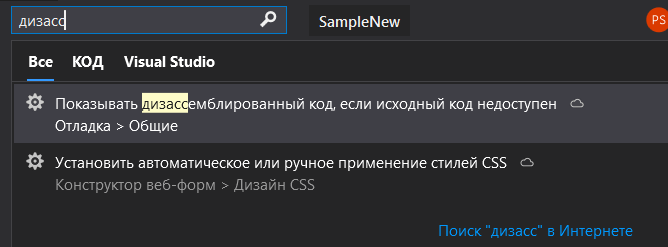
**Ответ**

Для начала я взял исполняемый модуль из 3-его лабораторного занятия (SampleNew.exe) и открыл его в Visual Studio 2019. (почему именно Visual Studio, потому что именно в этой среде разработки нам приходилось очень много работать на 1-ом и 2-ом курсах, а ещё в этой среде разработки есть очень хорошие средства отладки).

Для начала я методом перетаскивания перетащил SampleNew.exe в Visual Studio:

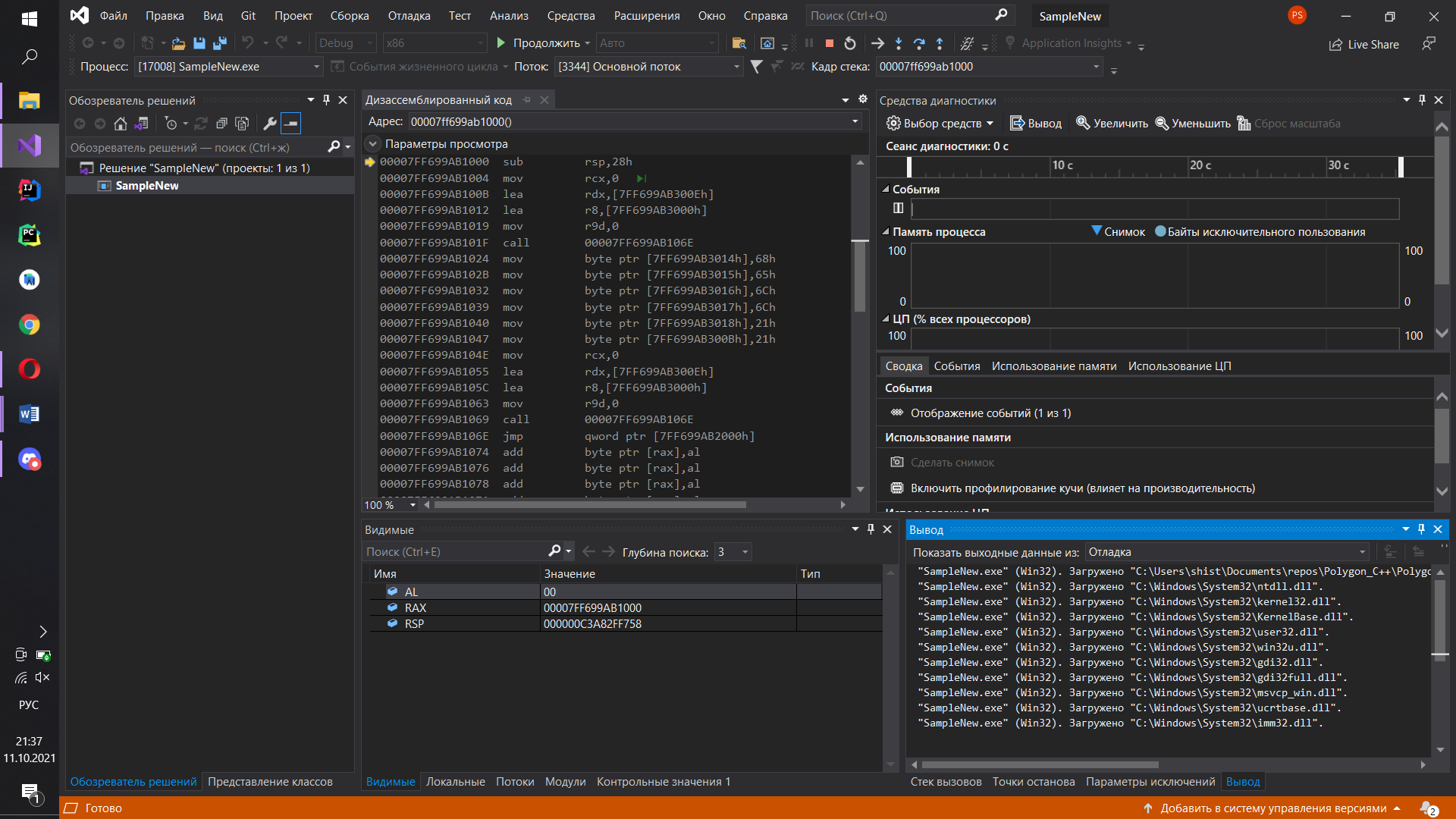


Затем я включил функцию показывания дизассемблированного кода, что нам пригодится при отладке исходного текста на языке assembler:

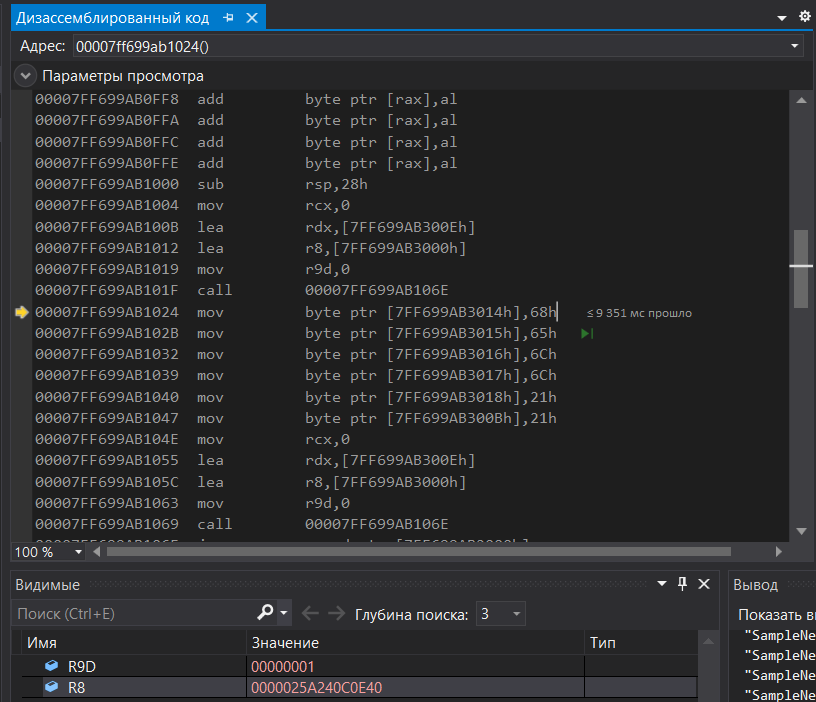




Затем я нажал F10 и начал пошаговую отладку:



Затем я прокликал клавишу F10 столько раз, чтобы добраться до интересующей нас строчки (хотя это было не обязательно):



Первая последовательность символов в нашей строке есть ничто иное, как **адрес команды**:



Таким образом, **адрес команды**: **00007FF699AB1024** (кстати говоря, эта штука хранится в регистре RIP)

**Шестнадцатеричное представление** этой команды **в памяти ЭВМ** будет выглядеть в виде 8-ми байтов:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 00 | 00 | 7F | F6 | 99 | AB | 10 | 24 |

**Содержимое регистра EIP, когда исходный текст будет оттранслирован с получением объектного модуля, который в свою очередь будет далее пролинчеван с получением на внешнем устройстве исполнимого модуля, который в свою очередь будучи загружен операционной системой в оперативную память превратится в программу, которой будет передано управление, и когда в конце концов дойдёт дело до исполнения этой команды,** будет представлять из себя совокупность последних 4-ёх байтов из 8-ми байтов шестнадцатеричного представления команды в памяти ЭМИ, т.е.:

**99AB1024** (кстати говоря, регистр EIP – это младшая часть регистра RIP)

Далее мы посмотрим на вот этот адрес:



По сути, это адрес области памяти для message+6, соответственно для получения адреса области памяти message мы должны отнять 6 от последнего числа (байта) в этом адресе.

Таким образом, **адрес области памяти, поименованной как message, в шестнадцатеричном виде** будет выглядеть так:

**7FF699AB3008**