**Код:** mov ax, @data

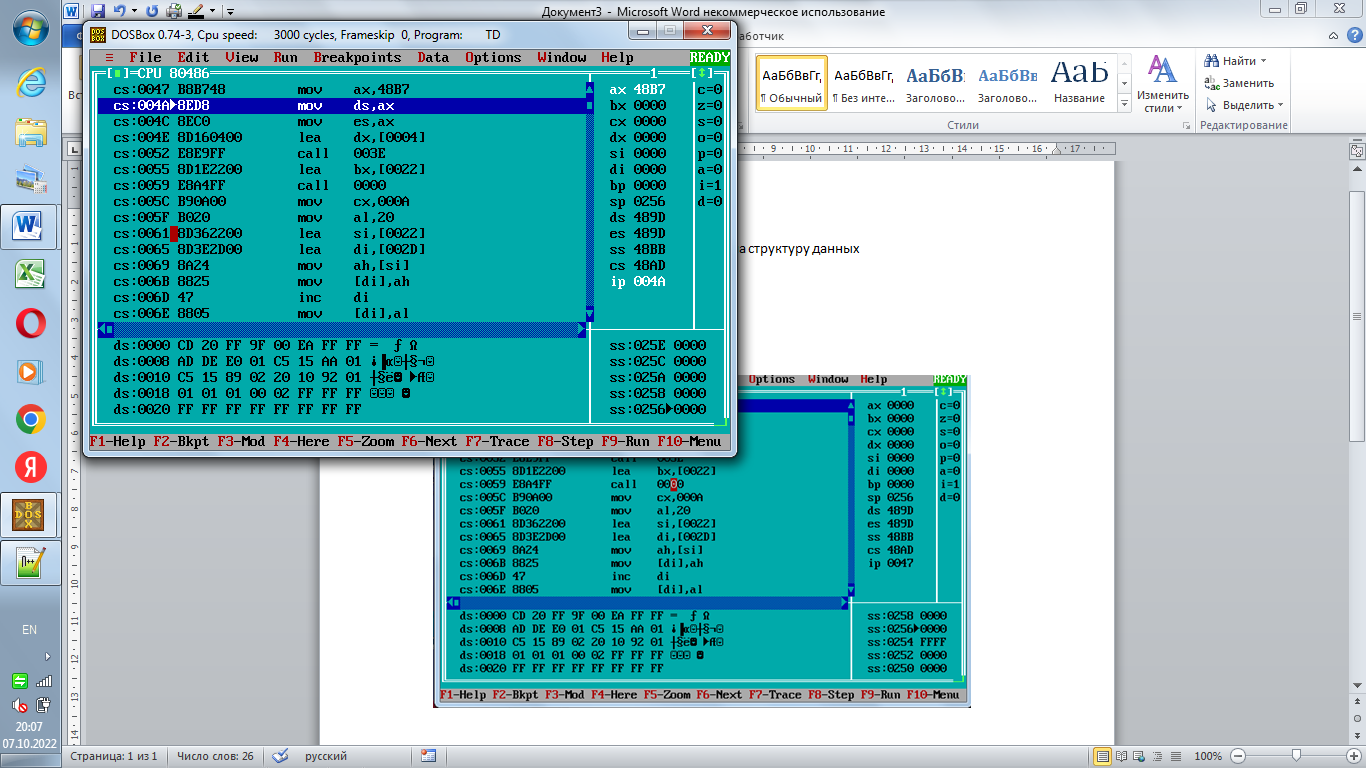
**Пояснения:** в регистр ax записываем указатель на структуру данных

**Изменения:**

- регистр ax изменится на 48B7

- регистр ip изменится на 004А

**Проверяем:**



**Код:** mov ds, ax ; load data segment

**Пояснения:** загружаем сегмент данных

П.С. Он выглядит у меня так:

message0 db 'String: $'

message1 db 0Ah, 0Dh, 'Begin: $'

message2 db 0Ah, 0Dh, 'Result: $'

FullData STRUC

one db 11 dup(0) ; тут мы выделяем 11 байт памяти и записываем туда символ 0 (потому что 0 имеет код 0 в кодировке ASCII)

two db 11 dup(0); тоже самое

FullData ENDS

FD FullData 1 dup(<>) ; выделяем место под 1 струтктуру. Так как структура и в ней размер полей задан, то пишем <>

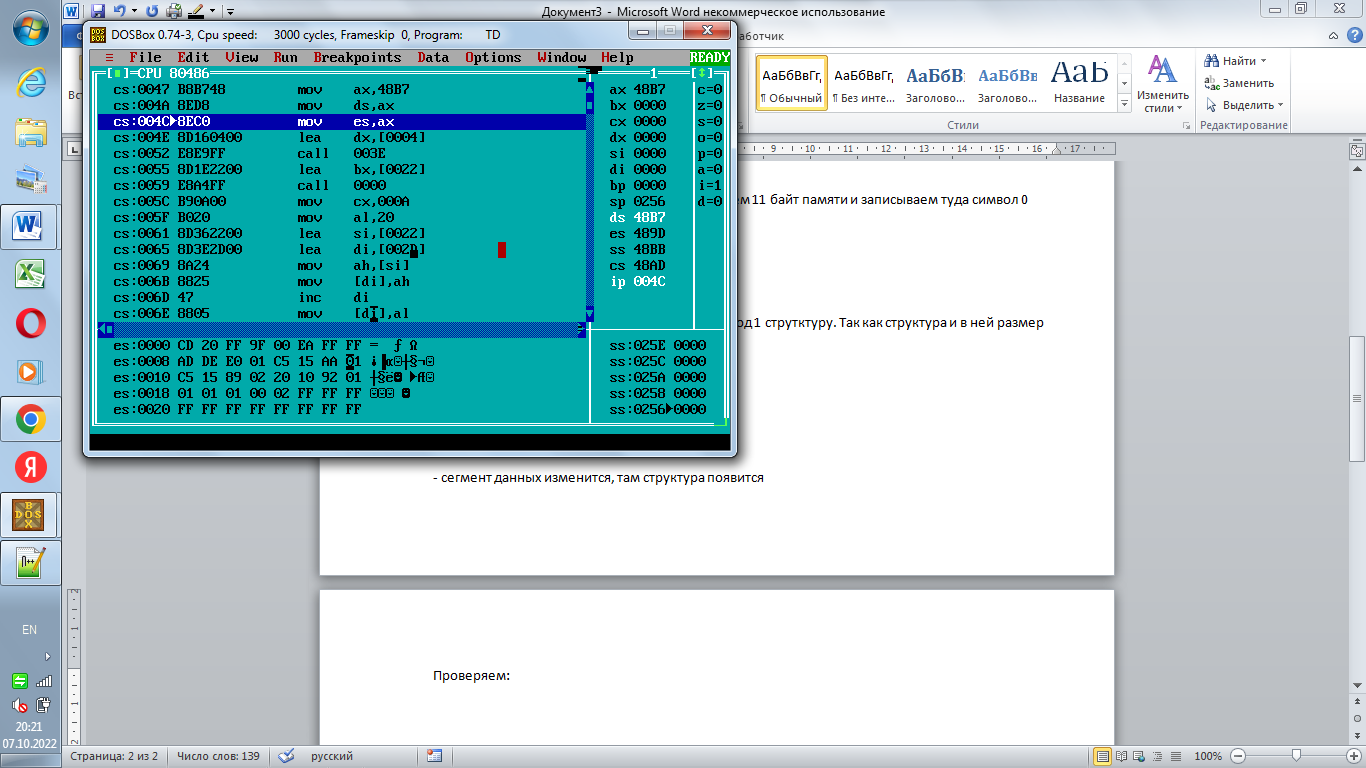
**Изменения:**

- регистр ds изменится на 48B7

- регистр ip изменится на 004C

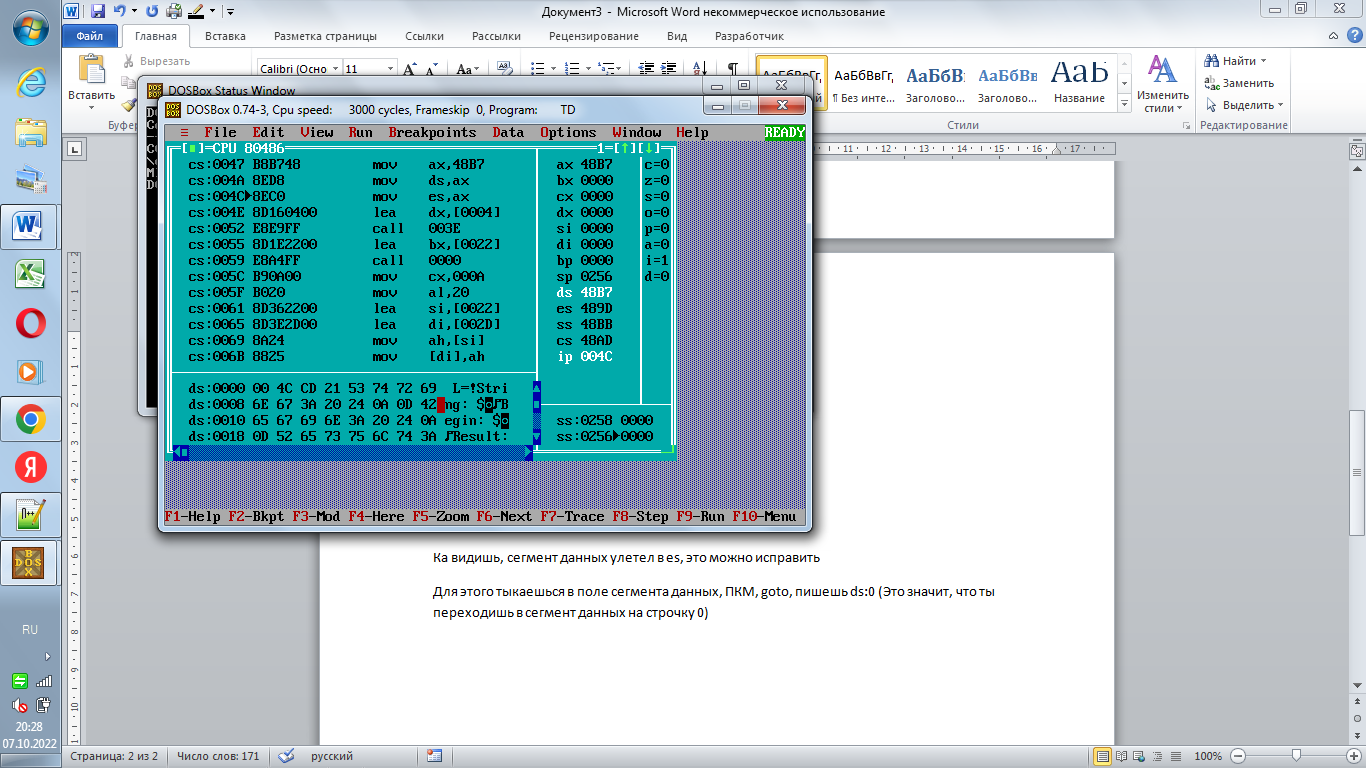
- сегмент данных изменится, там структура появится

**Проверяем:**



Ка видишь, сегмент данных улетел в es, это можно исправить

Для этого тыкаешься в поле сегмента данных, ПКМ, goto, пишешь ds:0 (Это значит, что ты переходишь в сегмент данных на строчку 0)



Все хорошо

**Код:** lea dx, message0

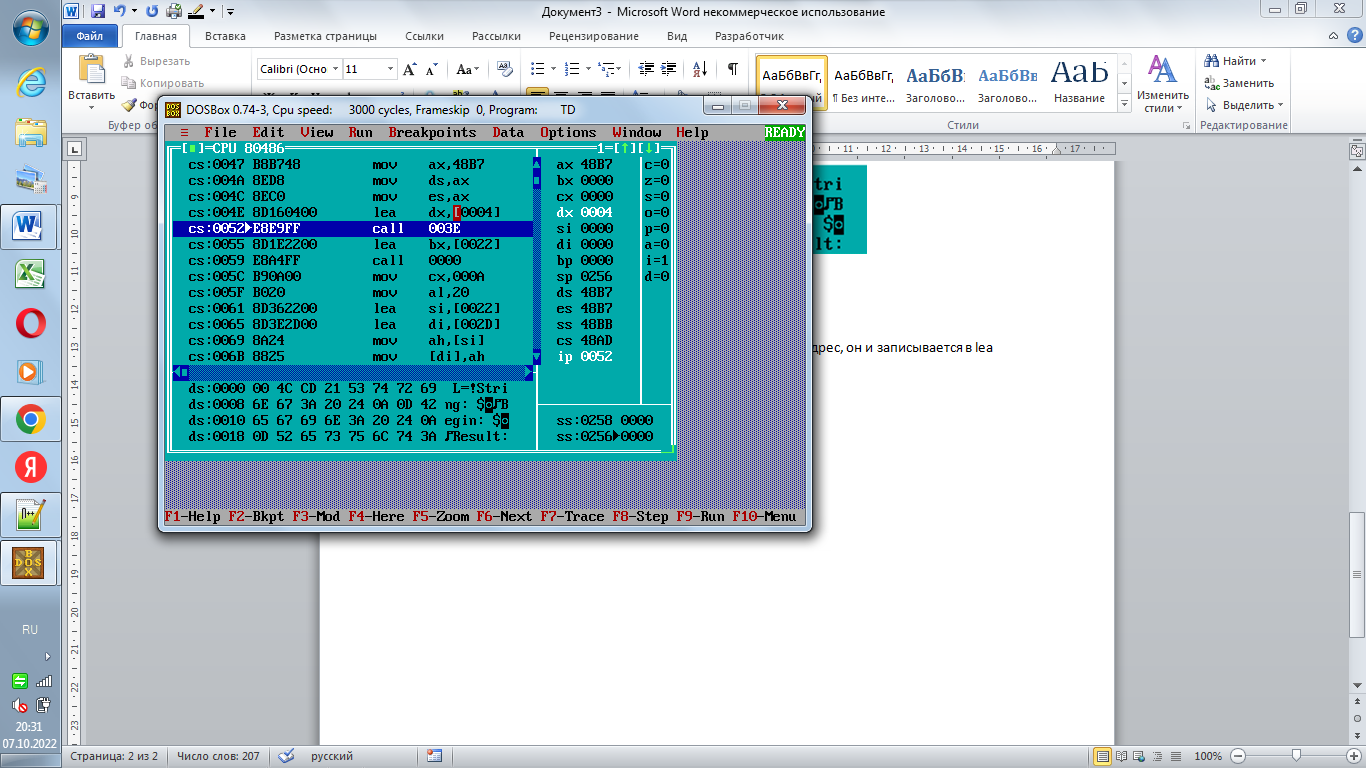
**Пояснения:** вывод строки String на экран, у нее тоже есить адрес, он и записывается в lea

**Измения:**

- регитср dx изменится на 0004

- регистр ip изменится на 0052

**Проверяем**



**Код:** call MESSAGE

**Пояснения:** вызываем подпрограмму

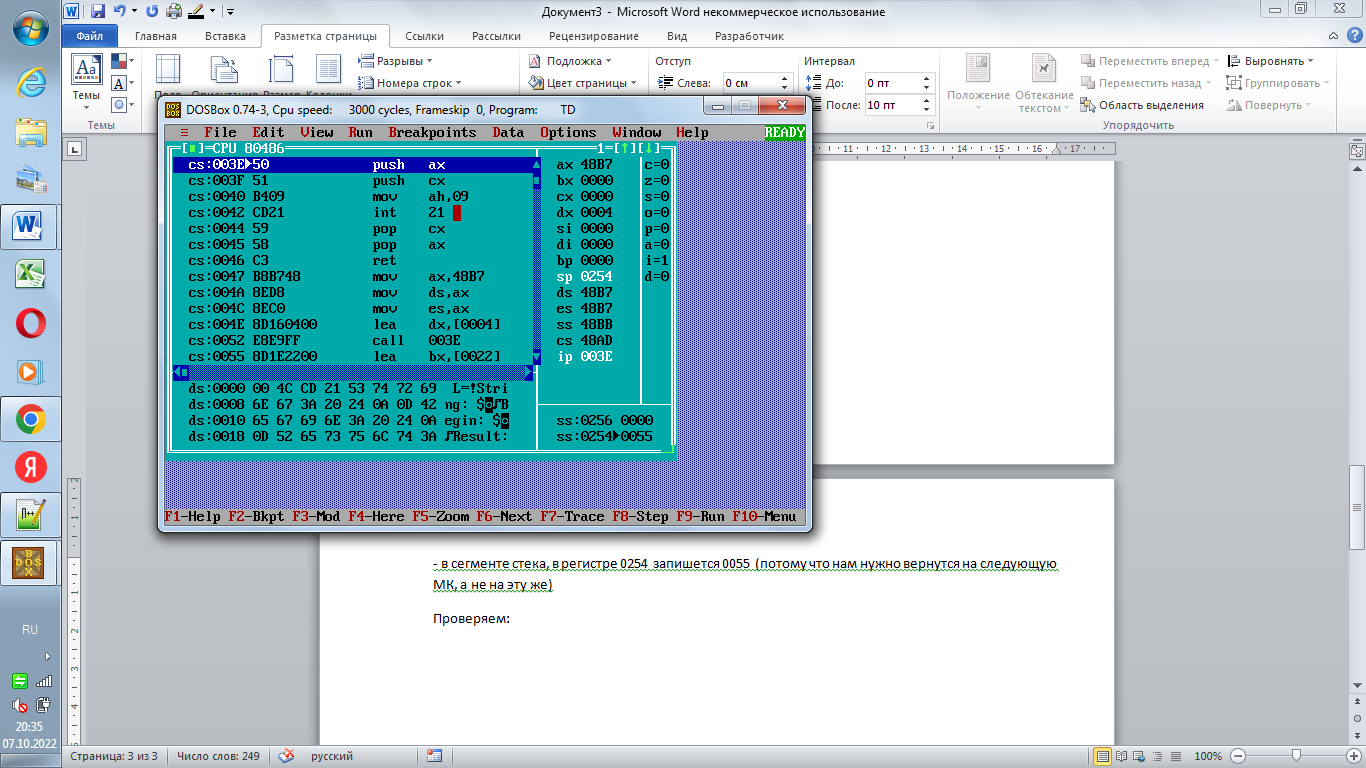
**Изменения:**

- регистр ip изменится на 003Е

- регистр sp изменится на 0254

- в сегменте стека, в регистре 0254 запишется 0055 (потому что нам нужно вернутся на следующую МК, а не на эту же)

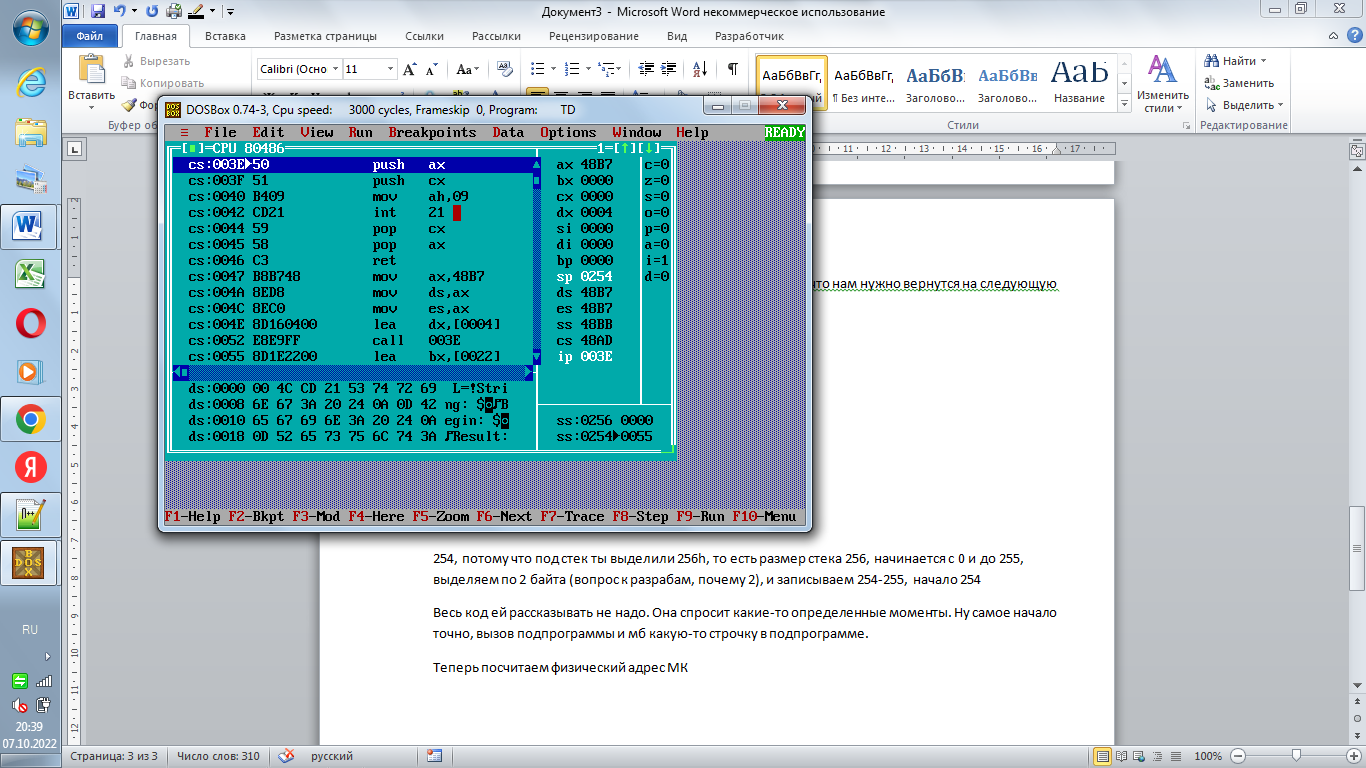
**Проверяем:**



254, потому что под стек ты выделили 256h, то есть размер стека 256, начинается с 0 и до 255, выделяем по 2 байта (вопрос к разрабам, почему 2), и записываем 254-255, начало 254

Весь код ей рассказывать не надо. Она спросит какие-то определенные моменты. Ну самое начало точно, вызов подпрограммы и мб какую-то строчку в подпрограмме.

**Теперь посчитаем физический адрес МК**



CS: 48AD

Дописываем 0 🡪 48АD0

Прибавляем Ip 🡪48AD0+003E= (4+0) (8+0)(A+0) (D(14)+3) (0+E) = 4 8 17 E = 48(A+1переноса) 1 E

= 48B1E