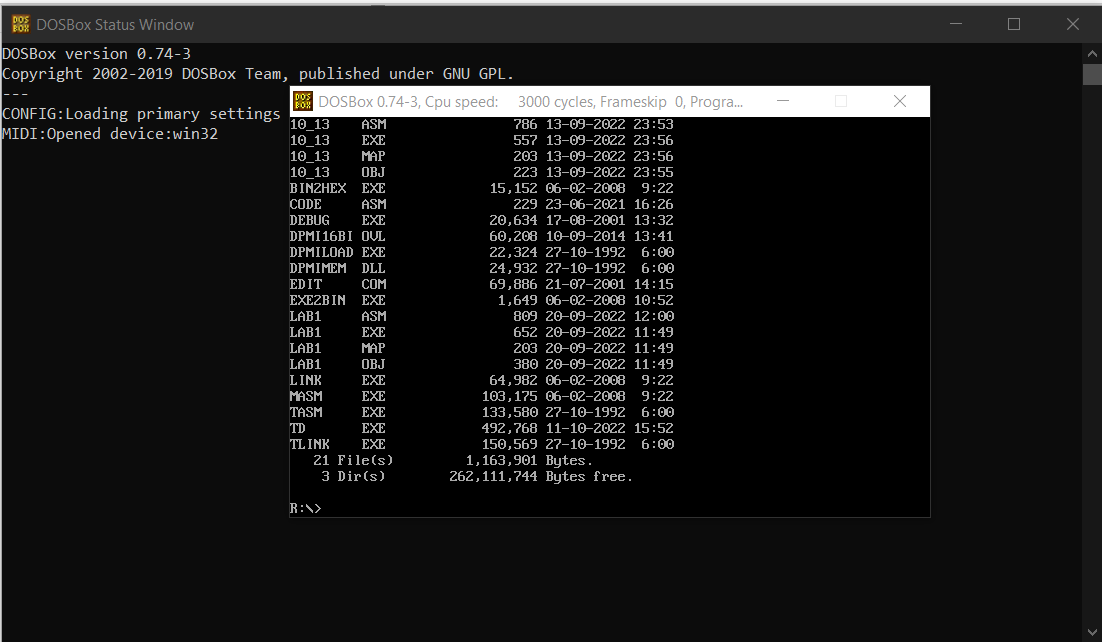
Лабораторное занятие №6 11.10.2022

Доскоч Роман ФПМИ 4 курс 13 группа

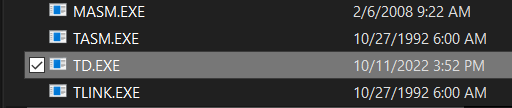
Условие на лабораторное занятие №6 до следующего вторника (18.10.2022)

1. «Работаем» в рамках DosBox.
2. Ищем программное средство отладки под названием td.exe.
3. Получаем исполнимые модули типа com и exe  —  для какой угодно задачи — какая Вам нравится (можно из Лабы 1 или Лабы 2).
4. Проверяем с помощью отладчика, действительно ли сегментные регистры получают указанные в Лекции  и в книге про ассемблер значения. Находим в программе самую длинную и самую короткую КОМАНДЫ. Находим в регистрах самое большое значение и переводим в 10-ую систему счисления без калькулятора.
5. К понравившейся Вам задаче добавляем подзадачу ВЫВОДА НА ЭКРАН в 16-ичной системе счисления ЗНАЧЕНИЙ, находящихся в сегментных регистрах НА МОМЕНТ ПРЕВРАЩЕНИЯ ИСПОЛНИМОГО МОДУЛЯ в программу. И сравниваем полученное значение с тем, что есть в отладчике. Перевод в 16-ую систему счисления значения, сидящего в AX, есть в решении задачи по выводу на экран изображений символов кодировочной таблицы.
6. Осуществляем отладку и ПОШАГОВОЕ ВЫПОЛНЕНИЕ полученной программы по выводу на экран значений содержимого сегментных регистров.
7. Пробуем с помощью отладчика td.exe пошагово выполнить программу, которая получается из исполнимого модуля Sample.exe. Пытаемся определить, какие значения получают сегментные регистры. Понятно, что ничего не получится. Обманул? Качаем из помойки какой угодно другой отладчик-дизассемблер, который «фурычит» на 64-разрядных ЭВМ.  И добавляем к боксику фрагмент, выводящий на экран значения в 16-ичной системе счисления ВСЕХ РЕГИСТРОВ, использованных в тексте Боксика!!!!
8. С помощью скачанного из помойки отладчика ПОШАГОВО выполняем Боксик (НЕ СКАЧУЩИЙ)!
9. Результат — три файла — исходные тексты, исполнимые модули и ОТЧЁТ о проделанной работе ПРИСЛАТЬ мне на почту К СЛЕДУЮЩЕМУ ВТОРНИКУ!!!!
10. Продолжаем «работать» над заданиями из занятия №5 и над Лабораторной работой №2. Если возникают какие-то вопросы — задавайте их письменно ПО ПОЧТЕ!!!!

1 «Работаем» в рамках DosBox



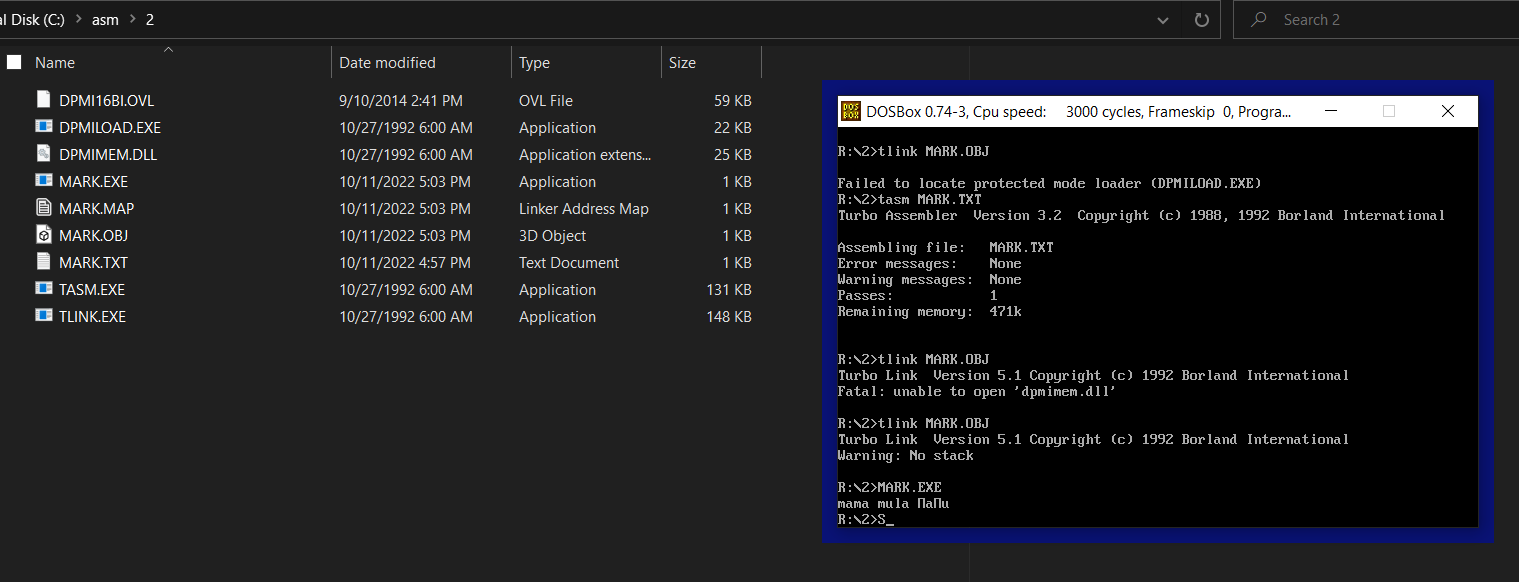
2 Ищем программное средство отладки под названием td.exe.



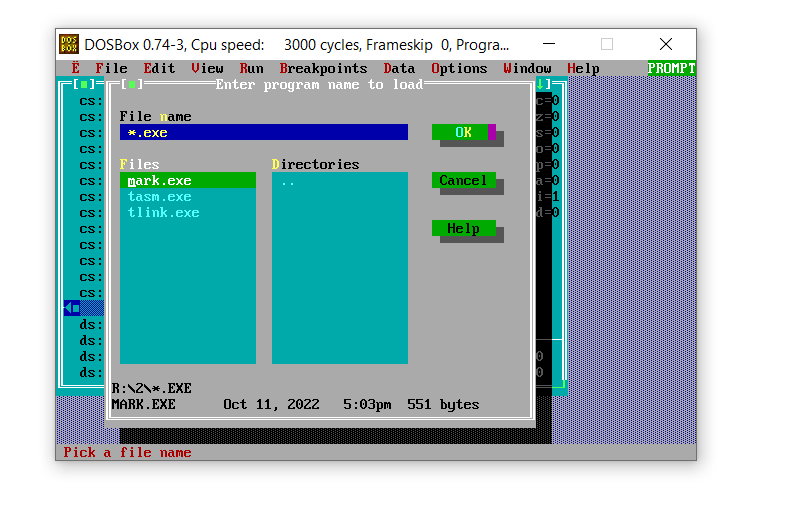
Нашел!

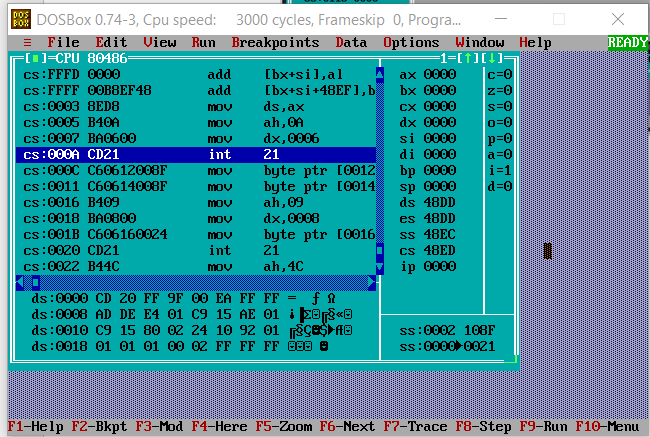
3 Получаем исполнимые модули типа com или exe  —  для какой угодно задачи — какая Вам нравится (можно из Лабы 1 или Лабы 2).

Выбираю задачу из 2 лабораторного занятия с выводом ***«MAMA MILA ПАПU».***

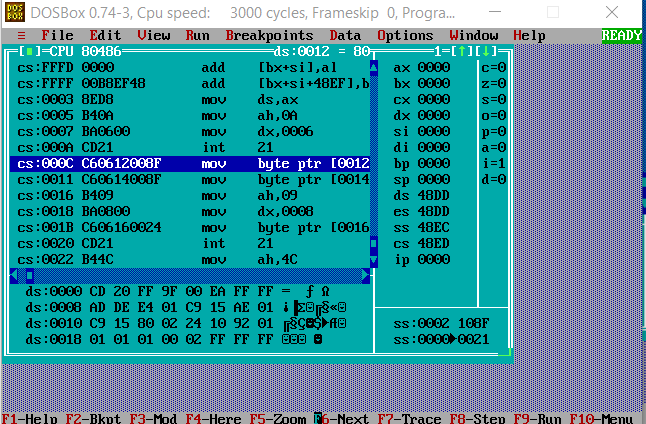


4 Проверяем с помощью отладчика, действительно ли сегментные регистры получают указанные в Лекции  и в книге про ассемблер значения. Находим в программе самую длинную и самую короткую КОМАНДЫ. Находим в регистрах самое большое значение и переводим в 10-ую систему счисления без калькулятора.

Загрузил в турбодебагер ехе

Самая маленькая команда 

И самая большая



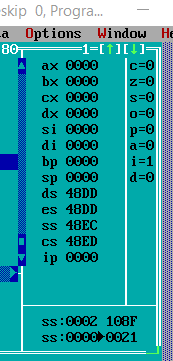
Самая большая

cs: 000C C60612008F mov byte ptr [0012], 8F

Самая маленькая

cs: 000A CD21 int 21

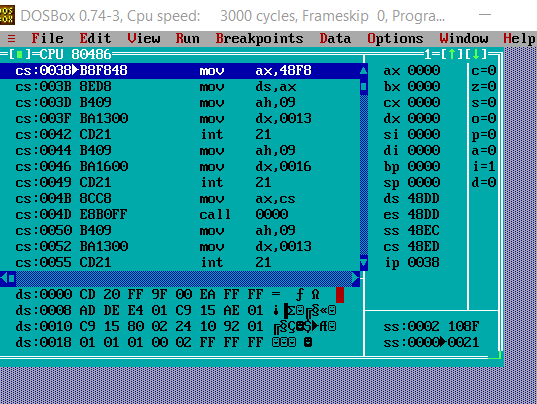
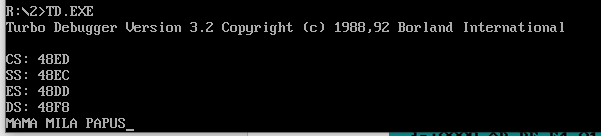
Самое большое значение в регистрах это 48ED



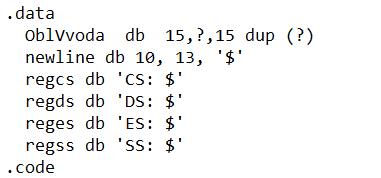
48ED₁₆ = 4 × 16³ + 8 × 16² + 14 × 16¹ + 13 × 16⁰ = 18669

5. К понравившейся Вам задаче добавляем подзадачу ВЫВОДА НА ЭКРАН в 16-ичной системе счисления ЗНАЧЕНИЙ, находящихся в сегментных регистрах НА МОМЕНТ ПРЕВРАЩЕНИЯ ИСПОЛНИМОГО МОДУЛЯ в программу. И сравниваем полученное значение с тем, что есть в отладчике. Перевод в 16-ую систему счисления значения, сидящего в AX, есть в решении задачи по выводу на экран изображений символов кодировочной таблицы.

В исходный текст моей выбранной задачи добавлю процедуру **print\_reg** для вывода регистров.

Всего сегментных регистров у нас  

В сегменту data я поместил шаблон для вывода наших регистров



Вот сам код вывода регистров

mov ah, 09h

lea dx, regcs

int 21h

mov ax, cs

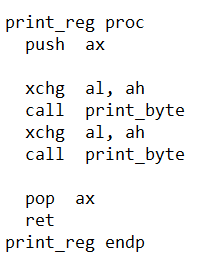
call print\_reg

mov ah, 09h

lea dx, newline

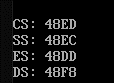
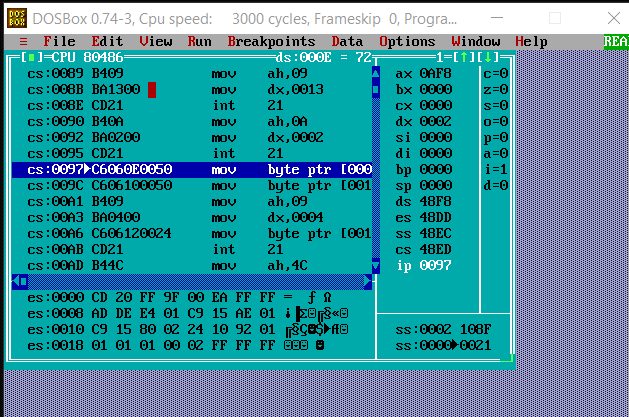
int 21h

и вот сама функция вывода регистров



Исходный текст находится в файле Mark.TXT

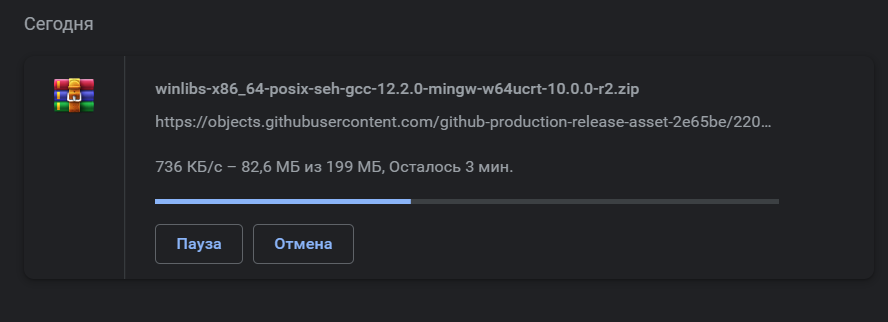
6. Осуществляем отладку и ПОШАГОВОЕ ВЫПОЛНЕНИЕ полученной программы по выводу на экран значений содержимого сегментных регистров.

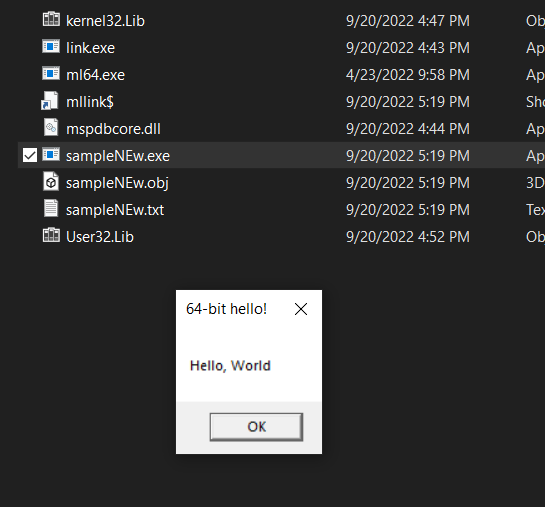
7. Пробуем с помощью отладчика td.exe пошагово выполнить программу, которая получается из исполнимого модуля Sample.exe. Пытаемся определить, какие значения получают сегментные регистры. Понятно, что ничего не получится. Обманул? Качаем из помойки какой угодно другой отладчик-дизассемблер, который «фурычит» на 64-разрядных ЭВМ.  И добавляем к боксику фрагмент, выводящий на экран значения в 16-ичной системе счисления ВСЕХ РЕГИСТРОВ, использованных в тексте Боксика!!!!

8. С помощью скачанного из помойки отладчика ПОШАГОВО выполняем Боксик (НЕ СКАЧУЩИЙ)!

Скачаем из помойки отладчик поторый фурычит на 64-ичной системе и это будет GDB



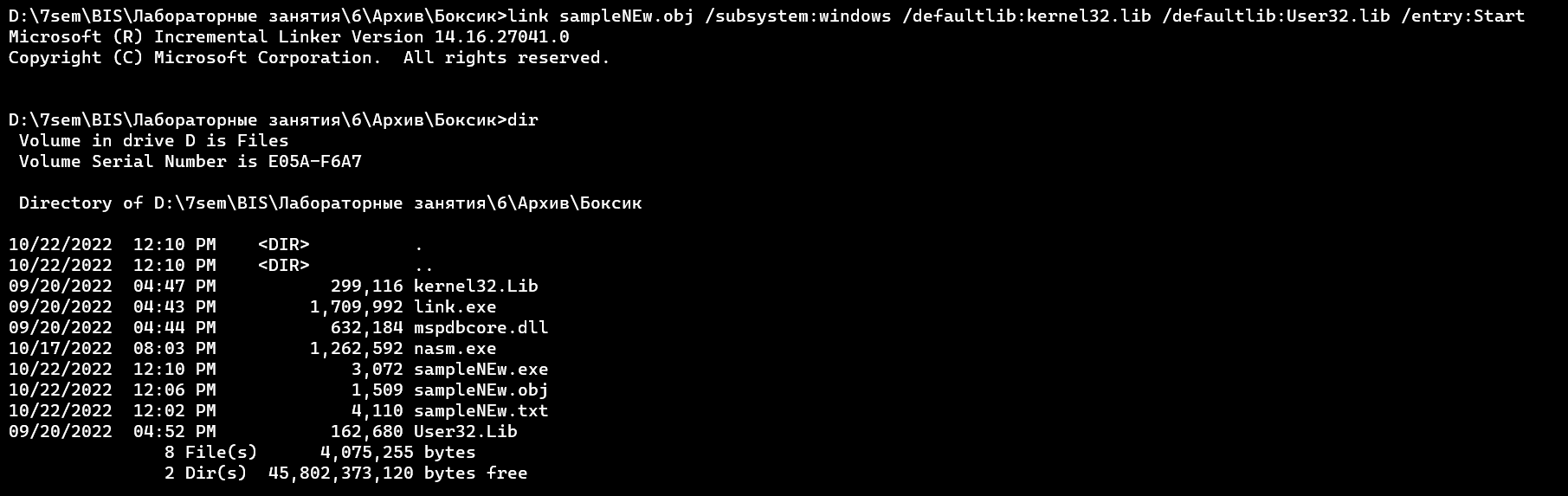
Запустим сначало просто Боксик

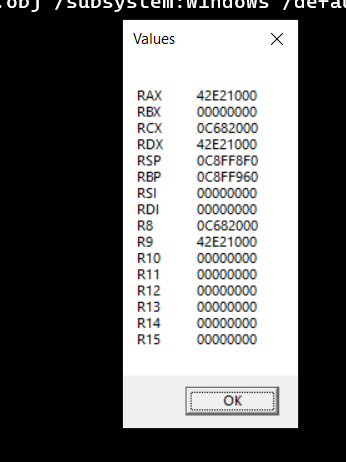


Изменим исходный текст так что бы выводились значения регистров

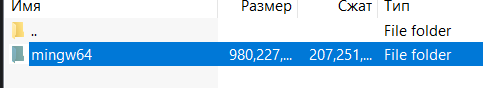
Начнем процесс создания исполнимого модуля



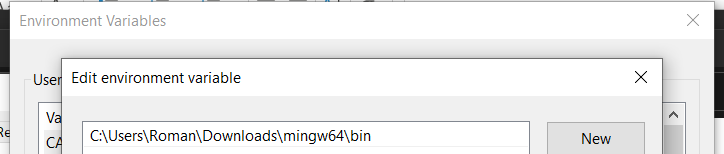




Далее установим GDB



Поместим папку bin в которой и находится наш GDB.EXE в глобальные переменные Path





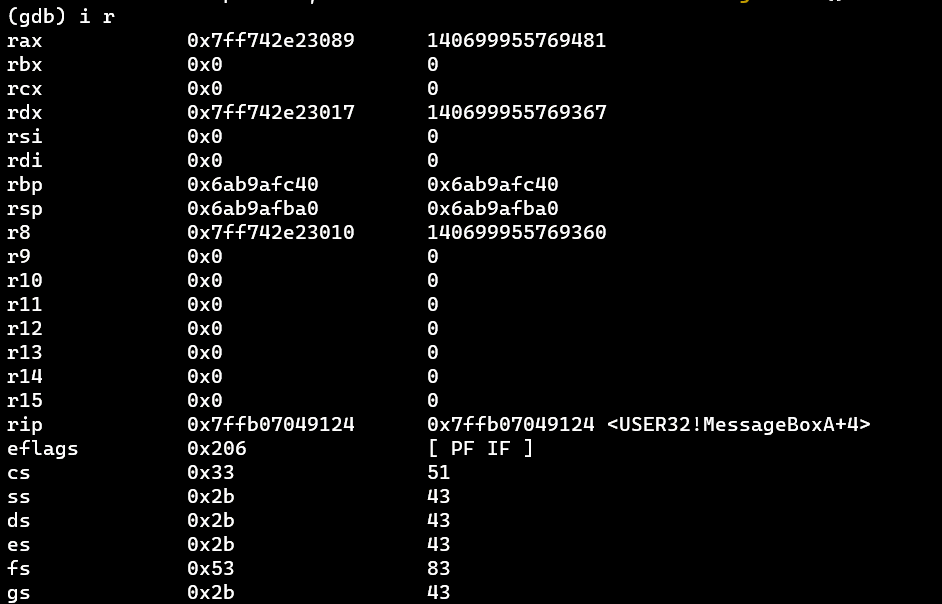
Запустим его с нашим исполняемым модулем



Поставим break point



Выведем вс. Информацию о регистрах info registers



C - continue