**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**



ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

**Лабораторная работа №1**

по дисциплине: Архитектура вычислительных систем

тема: «Разработка программ на ассемблере.

Работа с отладчиком x32dbg, пакетом masm32»

Выполнил: ст. группы ПВ-233

Мовчан Антон Юрьевич

Проверили:

ст. пр. Осипов Олег Васильевич

Белгород 2025 г.

**Лабораторная работа №1**

**Цель работы:** получить навыки создания простейших ассемблерных программ с использованием пакета masm32 и научиться пользоваться отладчиком x32dbg.

**Вариант 8**

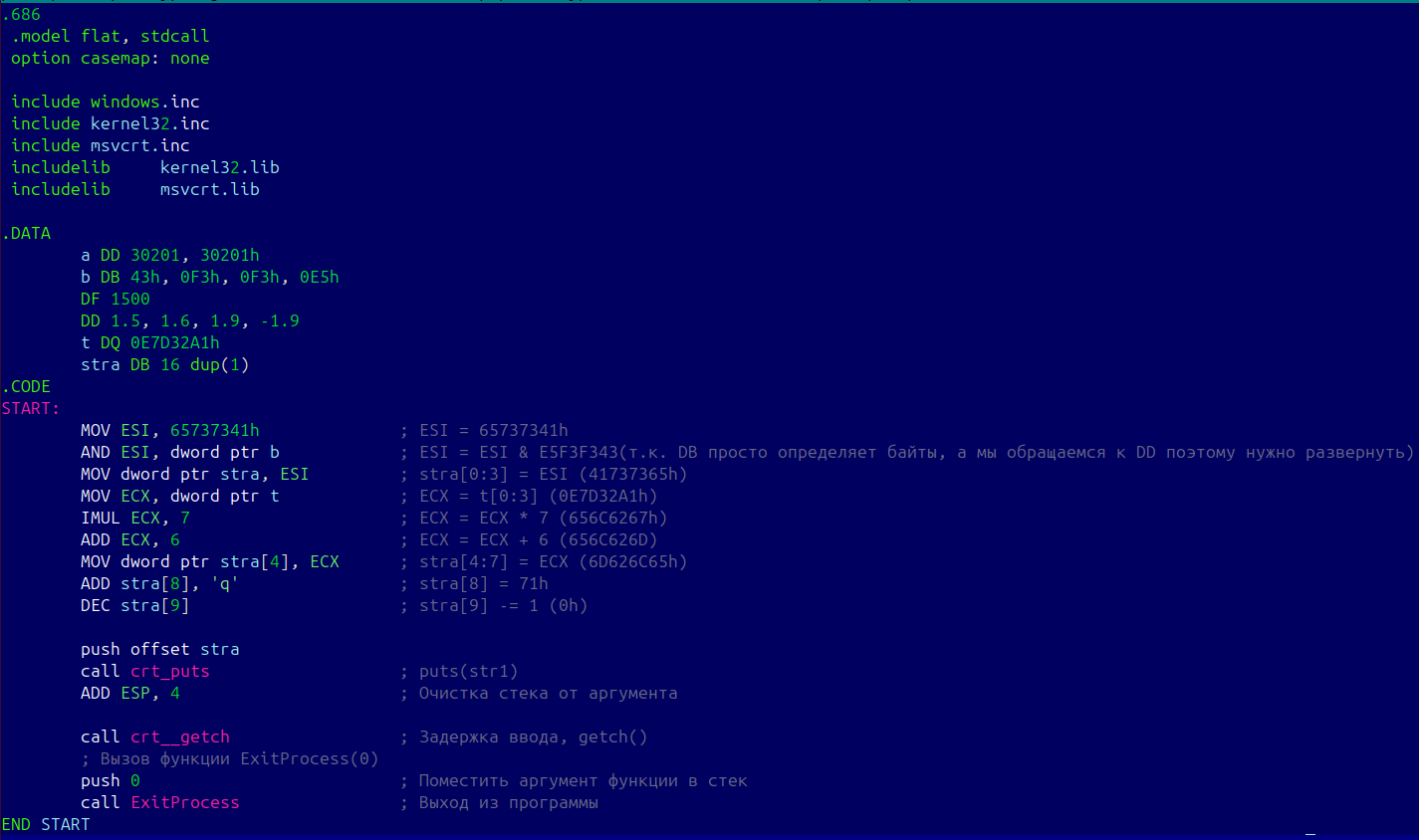
1. 1. Ознакомиться со средой x32dbg и компилятором masm32.
2. 2. Создать и скомпилировать программу в соответствии с вариантом задания. В программу включить комментарии с описанием, что делает каждая инструкция. Подробное описание каждой команды можно найти в приложении учебника В.И. Юрова «Assembler», начиная со стр. 511. Комментарии следует выровнять по левому краю (как в примере).
3. 3. С помощью отладчика определить местонахождение переменных, строк и массивов в сегменте данных, а также их размер. Составить таблицу и подробное описание ячеек сегмента данных (как в примере).
4. 4. Выполнить пошаговую трассировку программы. Определить какие регистры, флаги и ячейки памяти изменяют свои значения в процессе выполнения команд. Обеспечить корректное завершение программы вызовом системной функции ExitProcess с кодом завершения 0. Если в сегменте данных есть строки, то вывести её в консоль. Трассировку требуется выполнить до команды «call ExitProcess» включительно. Составить для каждой инструкции таблицу трассировки (как в примере).
5. 5. Сделать выводы о проделанной работе.

Сегменты данных и кода имеют следующее содержание:

1. .DATA
2. a DD 30201, 30201h
3. b DB 43h, 0F3h, 0F3h, 0E5h
4. DF 1500
5. DD 1.5, 1.6, 1.9, -1.9
6. t DQ 0E7D32A1h
7. stra DB 16 dup(1)
8. .CODE
9. START:
10. MOV ESI, 65737341h
11. AND ESI, dword ptr b
12. MOV dword ptr stra, ESI
13. MOV ECX, dword ptr t
14. IMUL ECX, 7
15. ADD ECX, 6
16. MOV dword ptr stra[4], ECX
17. ADD stra[8], 'q'
18. DEC stra[9]
19. END START

**Выполнение работы**

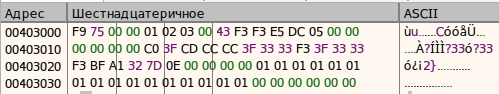
1. Создать файл lab1.asm со следующим содержанием:



2. Скомпилировать программу и получить исполняемый файл lab1.exe.

3. Открыть файл lab1.exe в отладчике.

4. Сегмент данных содержит три массива a, b, stra и переменную t



Адрес Шестанадцатеричное ASCII

00403000 F9 75 00 00 01 02 03 00 43 F3 F3 E5 DC 05 00 00 ùu......CóóåÜ...

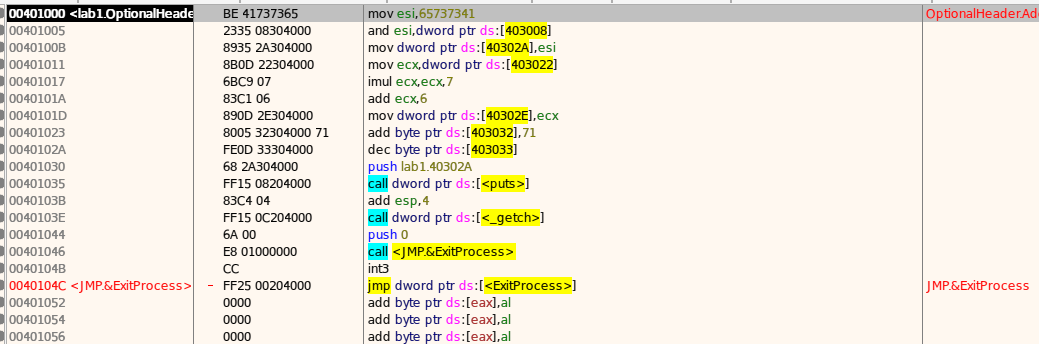
00403010 00 00 00 00 C0 3F CD CC CC 3F 33 33 F3 3F 33 33 ....À?ÍÌÌ?33ó?33

00403020 F3 BF A1 32 7D 0E 00 00 00 00 01 01 01 01 01 01 ó¿¡2}...........

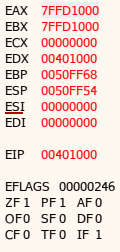
00403030 01 01 01 01 01 01 01 01 01 01 00 00 00 00 00 00 ................

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название перемен-ной | Начальный адрес | Конечный адрес | Размер данных, байт | Описание |
| a | 00403000 | 00403007 | 8 | Массив из двух 4-байтовых чисел |
| b | 00403008 | 0040300B | 4 | Массив из четырех 1-байтовых чисел |
| - | 0040300C | 00403011 | 6 | Неименовая область, содержащая число размером 6 байт |
| - | 00403012 | 00403021 | 16 | Неименовая область, содержащая четыре вещественных числа длиной 4 байт |
| t | 00403022 | 00403029 | 8 | 8-байтовое число |
| stra | 0040302A | 00403039 | 16 | Массив из 16 1-байтовых чисел |
| Общий размер сегмента данных: | | | 58 |  |

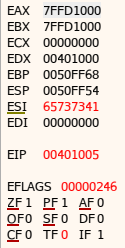
5. Пошаговая трассировка программы



Исходное состояние регистров:

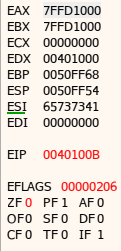






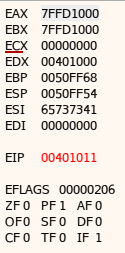
ESI = 65737341h





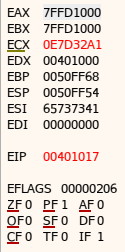
ESI = ESI & E5F3F343(т.к. DB просто определяет байты, а мы обращаемся к DD поэтому нужно развернуть)





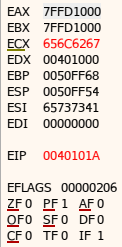
В начало stra записали содержимое регистра ESI





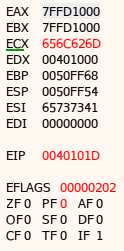
Записали в ECX значение t





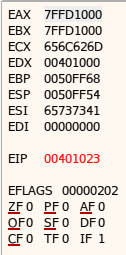
ECX = ECX \* 7 (656C6267h)





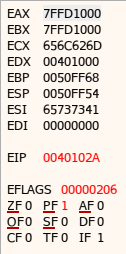
ECX = ECX + 6 (656C626D)





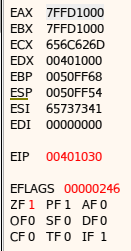
stra[4:7] = ECX (6D626C65h)





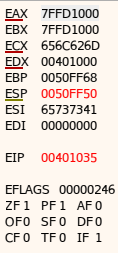
stra[8] = 71h (увеличили ячейку на 71)





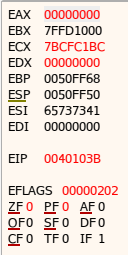
stra[9] -= 1 (0h)





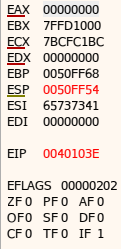
Кладет в стек значение из ячейки с адресом 40302А





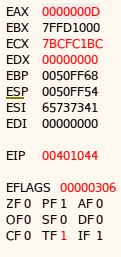
Вызов puts





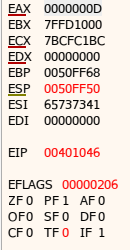
Вызов puts





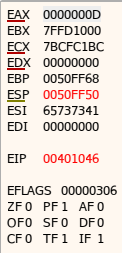
Вызов getch





Кладет в стек 0





Вызывает вызод из программы

Вывод: в ходе выполнения л.р. я получил навыки создания простейших ассемблерных программ с использованием пакета masm32 и научиться пользоваться отладчиком x32dbg.