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)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №1

по дисциплине: Архитектура вычислительных систем

тема: «Разработка программ на ассемблере.

Работа с отладчиком x32dbg, пакетом masm32»

Выполнил: ст. группы ПВ-223

Игнатьев Артур Олегович

Проверил:

Осипов Олег Васильевич

Белгород 2024 г.

# Содержание:

1. Название темы.
2. Цель работы.
3. Постановка задачи.
4. Вывод необходимых геометрических формул для построения изображения.
5. Реализации алгоритмов Брезенхейма для рисования отрезка и окружности.
6. Текст программы для рисования основных фигур.
7. Результат работы программы (снимки экрана).
8. Вывод о проделанной работе.

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

**Задачи**:

1. Ознакомиться со средой x32dbg и компилятором masm32.
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. 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. 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. Сделать выводы о проделанной работе.

**Задание варианта №3**

Сегменты данных и кода имеют следующее содержание:

.DATA

stra DB 20 DUP ('e')

DB 0

n DB 20 DUP (8)

a DW 500

b DD 0AB120001h, 100000

cc DQ 15.5, 15

d DD 7.5

.CODE

START:

MOV EAX, 03020100h

MOV EBX, DWORD PTR stra

ADD EBX, EAX

DEC stra[6]

MOV DWORD PTR stra, EBX

END START

Требуется определить местонахождение переменных, строк и массивов в сегменте данных, а также выполнить пошаговую трассировку программы.

**Выполнение работы**

1. Создать файл lab1.asm со следующим содержанием:

.DATA

stra DB 20 DUP ('e')

DB 0

n DB 20 DUP (8)

a DW 500

b DD 0AB120001h, 100000

cc DQ 15.5, 15

d DD 7.5

.CODE

START:

MOV EAX, 03020100h

MOV EBX, DWORD PTR stra

ADD EBX, EAX

DEC stra[6]

MOV DWORD PTR stra, EBX

END START

1. Скомпилировать программу и получить исполняемый файл lab1.exe.
2. Открыть файл lab1.exe в отладчике.
3. Сегмент данных содержит 1 - байтовую строку stra, 1 - байтовую неименованную переменную, 1 – байтовую n, 2 - байтовую a, 4 - байтовую b, 8 – байтовый массив cc и 4-байтовое d:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название перемен-ной | | Начальный адрес | Конечный адрес | Размер данных, байт | Описание |
| **stra** | | 00403000 | 00403013 | 20 | строка «eeeeeeeeeeeeeeeeeeee» |
|  | | 00403014 | 00403014 | 1 | однобайтовое число 0 |
| **n** | | 00403015 | 00403028 | 20 | строка «88888888888888888888» |
| **a** | | 004030F4 | 004030F4 | 2 | двухбайтное целое число 500 |
| **b** | | 0040302A | 0040302B | 8 | массив из 2-х четырёхбайтовых чисел содержащий 0AB120001h и 100000 |
| **cc** | 0040302D | | 0040302F | 16 | массив из 2 - х элементов по 8 байт, содержащий числа с плавающей точкой 15.5 и 15 |
| **d** | 00403030 | | 00403031 | 4 | переменная размером 4 байта, содержащая число с плавающей точкой 7.5 |
| Общий размер сегмента данных: | | | | **71** |  |

stra – строка символов.

Неименованная переменная со значением 0

n – строка символов.

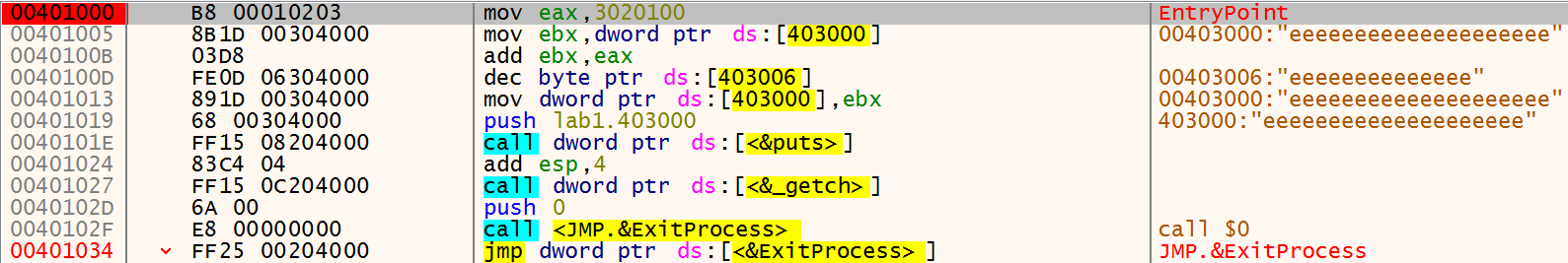
a – переменная со значением 500=1F416

b – массив из 2-х значений 0AB120001h и 100000

cc – массив из 2-х значений 15.5 и 15

s – переменная со значением 7.5

1. Пошаговая трассировка программы

****

Исходное состояние регистров:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EAX= | 00000000 | | EBX= | 00000000 | | ECX= | 24A80000 | EDX= | 00000000 |
| ESP= | 0019F7F4 | | EBP= | 0019F820 | | ESI= | 005F1F18 | EDI= | 00288000 |
| EIP= | 77EC8958 | |  | | | | | | |
| ZF= | 1 | PF= | 1 | AF= | 0 |  | | | |
| OF= | 0 | SF= | 0 | DF= | 0 |
| CF= | 0 | TF= | 0 | IF= | 1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MOV EAX, 03020100h** | | | | | | КОП: | B8 00010203 | | |
| EAX= | 0019FFCC | | EBX= | 00288000 | | ECX= | 00401000 | EDX= | 00401000 |
| ESP= | 0019FF78 | | EBP= | 0019FF84 | | ESI= | 00401000 | EDI= | 00401000 |
| EIP= | 00401000 | |  | | | | | | |
| ZF= | 1 | PF= | 1 | AF= | 0 |  | | | |
| OF= | 0 | SF= | 0 | DF= | 0 |
| CF= | 0 | TF= | 0 | IF= | 1 |
| Загружает в EAX значение 03020100h | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MOV EBX, DWORD PTR stra** | | | | | | КОП: | 8B1D 00304000 | | |
| EAX= | 03020100 | | EBX= | 00288000 | | ECX= | 00401000 | EDX= | 00401000 |
| ESP= | 0019FF78 | | EBP= | 0019FF84 | | ESI= | 00401000 | EDI= | 00401000 |
| EIP= | 00401005 | |  | | | | | | |
| ZF= | 1 | PF= | 1 | AF= | 0 |  | | | |
| OF= | 0 | SF= | 0 | DF= | 0 |
| CF= | 0 | TF= | 0 | IF= | 1 |
| Загружает адрес stra в EBX | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ADD EBX, EAX** | | | | | | КОП: | 03D8 | | |
| EAX= | 03020100 | | EBX= | 65656565 | | ECX= | 00401000 | EDX= | 00401000 |
| ESP= | 0019FF78 | | EBP= | 0019FF84 | | ESI= | 00401000 | EDI= | 00401000 |
| EIP= | 0040100B | |  | | | | | | |
| ZF= | 1 | PF= | 1 | AF= | 0 |  | | | |
| OF= | 0 | SF= | 0 | DF= | 0 |
| CF= | 0 | TF= | 0 | IF= | 1 |
| EBX изменяется на результат сложения с EAX | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DEC stra[6]** | | | | | | КОП: | FE0D 06304000 | | |
| EAX= | 03020100 | | EBX= | 68676665 | | ECX= | 00401000 | EDX= | 00401000 |
| ESP= | 0019FF78 | | EBP= | 0019FF84 | | ESI= | 00401000 | EDI= | 00401000 |
| EIP= | 0040100D | |  | | | | | | |
| ZF= | 0 | PF= | 1 | AF= | 0 |  | | | |
| OF= | 0 | SF= | 0 | DF= | 0 |
| CF= | 0 | TF= | 0 | IF= | 1 |
| Уменьшает на единицу значение седьмого байта в строке stra, изменяя 'e' на 'd' | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MOV DWORD PTR stra, EBX** | | | | | | КОП: | 891D 00304000 | | |
| EAX= | 03020100 | | EBX= | 68676665 | | ECX= | 00401000 | EDX= | 00401000 |
| ESP= | 0019FF78 | | EBP= | 0019FF84 | | ESI= | 00401000 | EDI= | 00401000 |
| EIP= | 00401013 | |  | | | | | | |
| ZF= | 0 | PF= | 0 | AF= | 0 |  | | | |
| OF= | 0 | SF= | 0 | DF= | 0 |
| CF= | 0 | TF= | 0 | IF= | 1 |
| Записывает значение EBX в первые четыре байта строки stra | | | | | | | | | |

**Вывод:** в ходе работы получены навыки создания простейших ассемблерных программ с использованием пакета masm32. Получен навык использования отладчика x32dbg.