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

Дмитриев Андрей Александрович

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 №2**

Сегменты данных и кода имеют следующее содержание:

.DATA

    strt DB "Some string", 0

    DW 400

    a DF 900

    mas DD 4\*8 DUP (3)

    s DQ 15.7

.CODE

START:

    MOV EAX, mas[0]

    MOV EBX, DWORD PTR [a]

    SUB EBX, EAX

    INC EBX

    MOV mas[4], 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:

.686

 .model flat, stdcall

 option casemap: none

.DATA

    strt DB "Some string", 0

    DW 400

    a DF 900

    mas DD 4\*8 DUP (3)

    s DQ 15.7

.CODE

START:

    MOV EAX, mas[0]

    MOV EBX, DWORD PTR [a]

    SUB EBX, EAX

    INC EBX

    MOV mas[4], EBX

END START

1. Скомпилировать программу и получить исполняемый файл lab1.exe.
2. Открыть файл lab1.exe в отладчике.
3. Сегмент данных содержит одну строку strt, одну неименованную переменную, 6-байтовую a, массив mas и 8-байтовое s:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название перемен-ной | Начальный адрес | Конечный адрес | Размер данных, байт | Описание |
| **strt** | 00403000 | 0040300B | 11 | строка «Some string» |
|  | 0040300С | 0040300D | 2 | двухбайтное целое число 400 |
| **a** | 0040300E | 00403013 | 6 | шестибайтовое целое число 900 |
| **mas** | 00403014 | 00403093 | 128 | массив из 32-х четырёхбайтовых целых чисел 3 |
| **s** | 00403094 | 0040309B | 8 | 8-байтовое вещественное число 15.7 |
| Общий размер сегмента данных: | | | **157** |  |

strt – строка символов, завершаемая 0-символом.

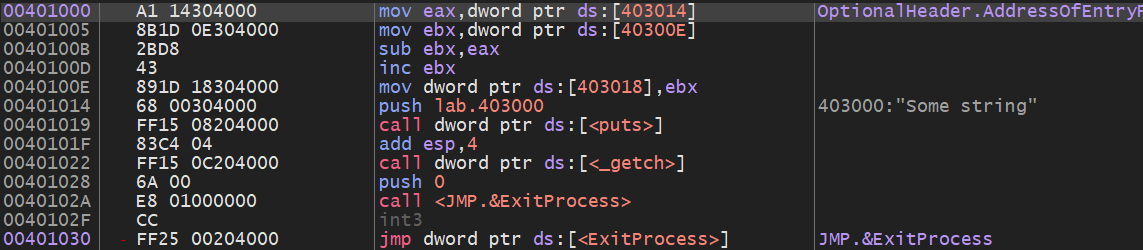
Неименованная переменная со значением 400=19016

a – переменная со значением 900=38416

mas – массив из 32-х значений инициированных 3

s – вещественное число 402F66…

1. Пошаговая трассировка программы

****

Исходное состояние регистров:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EAX= | 0019FFCC | | EBX= | 0023A000 | | 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 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **mov eax,dword ptr ds:[402014]** | | | | | | КОП: | A1 14204000 | | |
| EAX= | 00000003 | | EBX= | 0023A000 | | 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 |
| в ЕAX записываем 316 | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **mov ebx,dword ptr ds:[40200E]** | | | | | | КОП: | 8B1D 0E204000 | | |
| EAX= | 00000003 | | EBX= | 00000384 | | 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 |
| в ЕBX записываем 38416 | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **sub ebx,eax** | | | | | | КОП: | 2BD8 | | |
| EAX= | 00000003 | | EBX= | 00000381 | | EAX= | 00000003 | EDX= |  |
| ESP= | 0019FF78 | | EBP= | 0019FF84 | | ESP= | 0019FF78 | EDI= |  |
| EIP= | 0040100D | |  | | | | | | |
| ZF= | 0 | PF= | 1 | ZF= | 1 |  | | | |
| OF= | 0 | SF= | 0 | OF= | 0 |
| CF= | 0 | TF= | 0 | CF= | 0 |
| вычитаем из EBX значение ЕAX и записываем ответ 38116 в EBX | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **inc ebx** | | | | | | КОП: | 43 | | |
| EAX= | 00000003 | | EBX= | 00000382 | | EAX= | 00000003 | EDX= |  |
| ESP= | 0019FF78 | | EBP= | 0019FF84 | | ESP= | 0019FF78 | EDI= |  |
| EIP= | 0040100E | |  | | | | | | |
| ZF= | 0 | PF= | 1 | ZF= | 0 |  | | | |
| OF= | 0 | SF= | 0 | OF= | 0 |
| CF= | 0 | TF= | 0 | CF= | 0 |
| инкрементируем значение ЕBX и получаем 38216 в том же регистре | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **mov dword ptr ds:[40200E], ebx** | | | | | | КОП: | 891D 18204000 | | |
| EAX= | 00000003 | | EBX= | 00000382 | | EAX= | 00000003 | EDX= |  |
| ESP= | 0019FF78 | | EBP= | 0019FF84 | | ESP= | 0019FF78 | EDI= |  |
| EIP= | 00401014 | |  | | | | | | |
| ZF= | 0 | PF= | 1 | ZF= | 0 |  | | | |
| OF= | 0 | SF= | 0 | OF= | 0 |
| CF= | 0 | TF= | 0 | CF= | 0 |
| кладём в массив 38216 -> | | | | | | | | | |

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