*Федеральное государственное бюджетное образовательное учреждение* *высшего профессионального образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

**Отчет**

**по лабораторной работе № 1**

**Дисциплина:** Машинно-зависимые языки программирования

**Название лабораторной работы:**

Изучение среды и отладчика ассемблера

Студент гр. ИУ6-42  **\_\_\_\_\_\_29.03.2018 Бурлаков Александр Сергеевич**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2017

# ОСНОВНАЯ ЧАСТЬ

1. Запустите RADAsm, создайте файл проекта по шаблону консольного приложения. Внимательно изучите структуру программы и зафиксируйте текст с комментариями в отчете.

**Код шаблона программы:**

; Template for console application

.586

.MODEL flat, stdcall

OPTION CASEMAP:NONE

Include kernel32.inc

Include masm32.inc

IncludeLib kernel32.lib

IncludeLib masm32.lib

.CONST

MsgExit DB "Press Enter to Exit",0AH,0DH,0

.DATA

.DATA?

inbuf DB 100 DUP (?)

.CODE

Start:

;

; Add you statements

;

XOR EAX,EAX

Invoke StdOut,ADDR MsgExit

Invoke StdIn,ADDR inbuf,LengthOf inbuf

Invoke ExitProcess,0

End Start

2. Запустите шаблон на выполнение и просмотрите все полученные сообщения. Убедитесь, что текст программы и настройки среды не содержат ошибок.

**Результат выполнения шаблона программы:**



3. Добавьте директивы определения данных и команды сложения и вычитания.

.DATA

A SDWORD -30

B SDWORD 21

.DATA?

X SDWORD ?

Start:

mov EAX,A ; поместить число в регистр EAX

add EAX,5 ; сложить EAX и 5, результат в EAX

sub EAX,B ; вычесть B, результат в EAX

mov X,EAX ; сохранить результат в памяти

Найдите в отладчике внутреннее представление исходных данных, зафиксируйте его в отчете и поясните. Проследите в отладчике выполнение набранной вами программы и зафиксируйте в отчете результаты выполнения каждой добавленной команды (изменение регистров, флагов и полей данных).

**Внутреннее представление программы:**



Рисунок 1 – внутреннее представление данных

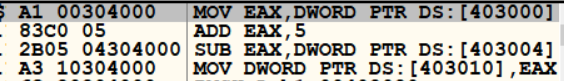


Рисунок 2 – внутреннее представление команд

Пояснение того, что делают команды

mov EAX, A – Значение регистра EAX изменилось на “FFFFFFE2”

add EAX, 5 – Значение регистра EAX изменилось на “FFFFFFE7”, флаг Z сменил значение на 0, а S на 1.

sub EAX, B – Значение регистра EAX изменилось на “FFFFFFD2”

mov X, EAX – Значение адреса “00403010” изменилось на “D2 FF FF FF”, ASCII – “Тяяя”

4. Введите следующие строки в раздел описания инициированных данных и определите с помощью отладчика внутренние представление этих данных в памяти. Результаты проанализируйте и занесите в отчет.

val1 BYTE 255

chart WORD 256

lue3 SWORD -128

alu BYTE ?

v5 BYTE 10h

BYTE 100101B

beta BYTE 23,23h,0ch

sdk BYTE “Hello”,0

min SWORD -32767

ar DWORD 12345678h

valar BYTE 5 DUP (1, 2, 8)

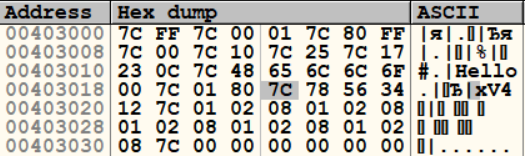


Рисунок 3 – внутреннее представление данных

Данные на рис.3 разделены вертикальными чертами “|”

Таблица 1 – Анализ внутреннего представления переменнных

|  |  |  |
| --- | --- | --- |
| Пер | Тип и значение | Описание представления переменной |
| val1 | BYTE 255 | FF – шестнадцатеричное представление числа 255 |
| chart | WORD 256 | 00 01 – запись во внутренней памяти означает число 0100h, или |
| lue3 | SWORD -128 | 80 FF – запись во внутренней памяти, означающее число FF80 или, переведя с помощью дополнительного кода, число -128 |
| alu | BYTE ? | В памяти записано как «00» |
| v5 | BYTE 10h | 10 – записано в шестнадцатеричной системе |
|  | BYTE 100101B | 25 – шестнадцатеричное представление числа 100101В |
| beta | BYTE 23, 23h, 0ch | 17 23 0С – переведенные в шестнадцатеричную систему числа |
| sdk | BYTE “Hello”, 0 | В левой таблице «ASCII» отображается строка «Hello», в таблице памяти представлен шестнадцатеричный код каждого символа |
| min | SWORD -32767 | 01 80 – смотри переменную lue3 |
| ar | DWORD 12345678h | 78 56 34 12 – запись во внутренней памяти, означающее число 12345678h |
| valar | 5 DUP (1, 2, 8) | 5 повторений чисел 01 02 08 |

5. Определите в памяти следующие данные: а) целое число 25 размером 2 байта со знаком; б) двойное слово, содержащее число -35; в) символьную строку, содержащую ваше имя (русскими буквами и латинскими буквами). Зафиксируйте в отчете внутреннее представление этих данных и дайте пояснение.



Рисунок 4 – внутреннее представление данных

А) целое число 25 размером 2 байта со знаком записано как «19 00» - переведенное в шестнадцатиричную систему число 25 и записанное в обратном порядке по 2 байта (было число 0019h – записали «19 00»)

Б) двойное слово, содержащее число -35 записано как «DD FF FF FF» - шестнадцатиричное число.

В) Строки, записанные в память отображаются в таблице ASCII справа. Слева приведен шестнадцатиричный код каждого символа.

6. Определите несколькими способами в программе числа, которые во внутреннем представлении (в отладчике) будут выглядеть как 25 00 и 00 25. Проверьте правильность ваших предположений, введя соответствующие строки в программу. Зафиксируйте результаты в отчете.

H32 SWORD 37

H4 SWORD 9472

H12 WORD -65499

H22 WORD -56064



Рисунок 5 – внутреннее представление данных

Пояснение:

, которое в память будет записано как «25 00»;

, которое в память будет записано как «00 25»;

, переведем его с помощью обратного кода: и, переведя это число в дополнительный код получим 25, а в память будет записано «25 00»;

по аналогии будет записано как «00 25»

7. Замените директивы описания знаковых данных на беззнаковые:

A DWORD -30

B DWORD 21

X DWORD ?

Запустите программу и прокомментируйте результат.

Пояснение того, что делают команды

mov EAX, A – Значение регистра EAX изменилось на “FFFFFFE2”

add EAX, 5 – Значение регистра EAX изменилось на “FFFFFFE7”, флаг Z сменил значение на 0, а S на 1.

sub EAX, B – Значение регистра EAX изменилось на “FFFFFFD2”

mov X, EAX – Значение адреса “00403010” изменилось на “D2 FF FF FF”, ASCII – “Тяяя”

8. Добавьте в программу переменную F1=65535 размером слово и переменную F2= 65535 размером двойное слово. Вставьте в программу команды сложения этих чисел с 1:

add F1,1

add F2,1

Проанализируйте и прокомментируйте в отчете полученный результат (обратите внимание на флаги).

Пояснение того, что делают команды

add F1, 1: значение флагов C и А сменилось на 1, значение в памяти изменилось с «FF FF» на «00 00».

add F2, 1: значение флага Z и C сменилось на 0, значение в памяти изменилось с «FF FF 00 00» на «00 00 01 00».

**ВЫВОД**

1. Выполнено знакомство со средой RADAsm, и с Masm32. Получены навыки работы с отладчиком.
2. Написаны программы, описываемые в заданиях и приведено объяснение тому, что эти программы делают.