

**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)**

Лабораторная работа № 1
дисциплина «Архитектура вычислительных систем»
по теме «Разработка программ на ассемблере. Работа с отладчиком OllyDbg и
пакетом masm32»

Выполнил: студент группы ВТ-31
Проверил:

Макаров Д.С
Осипов О.В.

Белгород 2019

Лабораторная работа № 1

«Разработка программ на ассемблере. Работа с отладчиком OllyDbg и пакетом masm32»

Цель работы: получить навыки создания и отладки простейших программ на ассемблере с использованием пакета masm32 и отладчика OllyDbg.

Вариант 9

Задание:

1. Ознакомиться со средой OllyDbg и компилятором masm32.
2. Создать и скомпилировать программу в соответствии с вариантом задания.
3. Отладить программу.
4. С помощью OllyDbg определить местонахождение переменных в сегменте данных, а также их размер.
5. Выполнить пошаговую трассировку программы. Определить какие регистры изменяют свои значения в процессе выполнения команд.

```
.386 ;Архитектура процессора
.model flat, stdcall ;Модель памяти и вызова подпрограмм.
option casemap: none ;Чувствительность к регистру
; Подключение библиотек и других необходимых файлов
include c:\masm32\include\windows.inc
include c:\masm32\include\kernel32.inc
include c:\masm32\include\user32.inc
includelib c:\masm32\lib\user32.lib
includelib c:\masm32\lib\kernel32.lib
; Сегмент данных
.DATA
    as DB 10, 13, "a-s", 0 ;массив из 6 однобайтовых чисел
    h DW 20 ;переменная размером с слово(2 байта) со значением 20 (0x14)
    w DW 100 ;переменная размером с слово со значением 100 (0x64)
    s DD ? ;неинициализированная переменная размером 4 байта
    DT 17834.55 ;переменная размером 10 байтов

.CODE

START:
    XOR EAX, EAX; обнуление регистра EAX
    MOV AX, h; копирование из данных из h в AX
    MUL w; умножение содержимого AX на переменную w
    MOV s, EAX; запись в переменную s содержимого регистра EAX
```

```

PUSH 0
CALL ExitProcess
END START

```

Ход работы

Сегмент данных содержит:

Address	Hex dump	ASCII
00402000	04 00 61 20 73 00 14 00	...a-s.%.
00402008	64 00 00 00 00 00 9A 99	d.....34
00402010	99 99 99 19 55 00 00 40	99994U%.0
00402018	00 00 00 00 00 00 00 00
00402020	00 00 00 00 00 00 00 00
00402028	00 00 00 00 00 00 00 00
00402030	00 00 00 00 00 00 00 00
00402038	00 00 00 00 00 00 00 00
00402040	00 00 00 00 00 00 00 00
00402048	00 00 00 00 00 00 00 00
00402050	00 00 00 00 00 00 00 00
00402058	00 00 00 00 00 00 00 00
00402060	00 00 00 00 00 00 00 00
00402068	00 00 00 00 00 00 00 00
00402070	00 00 00 00 00 00 00 00
00402078	00 00 00 00 00 00 00 00
00402080	00 00 00 00 00 00 00 00
00402088	00 00 00 00 00 00 00 00

Пошаговое выполнение программы

Состояние регистров до работы программы:

```

EAX 00000000
ECX 0012FFB0
EDX 7C90E514 ntdll.KiFastSystemCallRet
EBX 7FFDF000
ESP 0012FFC4
EBP 0012FFF0
ESI FFFFFFFF
EDI 7C910228 ntdll.7C910228
EIP 00401000 source.<ModuleEntryPoint>
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 1 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDF000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty -UNORM D1D8 01050104 00000000
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

```

XOR EAX, EAX: Код команды 33C0. Результат выполнения будет равен 0 при любом значении EAX, тем самым мы обнулим регистр.

```

EAX 00000000
ECX 0012FFB0
EDX 7C90E514 ntdll.KiFastSystemCallRet
EBX 7FFDB000
ESP 0012FFC4
EBP 0012FFF0
ESI FFFFFFFF
EDI 7C910228 ntdll.7C910228
EIP 00401000 source.<ModuleEntryPoint>
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 1 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDF000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty -UNORM D1D8 01050104 00000000
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

```

MOV AX,WORD PTR DS:[402006]: Код команды 66:A1 0620400. Команда MOV копирует данные из сегмента данных по указателю 402006 в младшие байты регистра EAX - AX. Результат работы операции:

```

EAX 00000014
ECX 0012FFB0
EDX 7C90E514 ntdll.KiFastSystemCallRet
EBX 7FFDB000
ESP 0012FFC4
EBP 0012FFF0
ESI FFFFFFFF
EDI 7C910228 ntdll.7C910228
EIP 00401008 source.00401008
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 1 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDF000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty -UNORM D1D8 01050104 00000000
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

```

MUL WORD PTR DS:[402008]: Код команды 66:F725 082040. Команда MUL берет 2 операнда, первый задается неявно и зависит от размера второго операнда. Регистр с результатом определяется кодом операции и размером множителей. В данном случае явный операнд размером WORD соответственно первым множителем является регистр AX а результат хранится в DX:AX.

```

EAX 000007D0
ECX 0012FFB0
EDX 7C900000 ntdll.7C900000
EBX 7FFDB000
ESP 0012FFC4
EBP 0012FFF0
ESI FFFFFFFF
EDI 7C910228 ntdll.7C910228
EIP 0040100F source.0040100F
C 0 ES 0023 32bit 0(FFFFFFFF)
P 0 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDF000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 0000202 (NO,NB,NE,A,NS,PO,GE,G)
ST0 empty -UNORM D1D8 01050104 00000000
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

```

MOV DWORD PTR DS:[40200A],EAX: Код команды A3:0A204000. Копирование данных из регистра EAX в сегмент данных по адресу 40200A.

Address	Hex dump	ASCII
00402000	0A 00 61 20 73 00 14 00	..a-s.т.
00402008	64 00 D0 07 00 00 9A 99	d.м...ъщ
00402010	99 99 99 19 55 8B 0D 40	щщщ+Uт.0
00402018	00 00 00 00 00 00 00 00
00402020	00 00 00 00 00 00 00 00
00402028	00 00 00 00 00 00 00 00
00402030	00 00 00 00 00 00 00 00
00402038	00 00 00 00 00 00 00 00
00402040	00 00 00 00 00 00 00 00
00402048	00 00 00 00 00 00 00 00
00402050	00 00 00 00 00 00 00 00
00402058	00 00 00 00 00 00 00 00
00402060	00 00 00 00 00 00 00 00

Приложение

Содержимое файла source.asm

```
.386
.model flat, stdcall
option casemap: none

include c:\masm32\include\windows.inc
include c:\masm32\include\kernel32.inc
include c:\masm32\include\user32.inc
includelib c:\masm32\lib\user32.lib
includelib c:\masm32\lib\kernel32.lib

.DATA
    as DB 10, 13, ;"a-s", 0
    h DW 20
    w DW 100
    s DD ?
    DT 17834.55

.CODE

START:
    XOR EAX, EAX
    MOV AX, h
    MUL w
    MOV s, EAX
    PUSH 0
    CALL ExitProcess
END START
```