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-211

Чувилко Илья Романович

Проверил:

Осипов Олег Васильевич

Белгород 2022 г.

Вариант 20

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

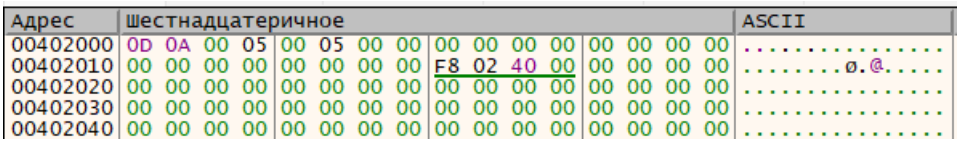
Выполнение работы:

1. Создать файл lab1.asm со следующим содержанием:

Код программы:

```
.686 ; Тип процессора
.model flat, stdcall ; Модель памяти и стиль вызова подпрограмм
option casemap: none ; Чувствительность к регистру
; --- Подключение файлов с кодом, макросами, константами, прототипами функций и т.д.
include windows.inc
include kernel32.inc
include user32.inc
include msvcrt.inc
; --- Подключаемые библиотеки ---
includelib user32.lib
includelib kernel32.lib
includelib msvcrt.lib
; --- Сегмент данных ---
.DATA
ente DB 13, 10, 0
a DW 5
b DW 5
result DD ?
p DF 17.5
ten DT 15.5
.CODE
START:
    XOR CX, CX
    MOV CX, b
    XOR AX, AX
    MOV AX, a
    MUL CX
    MOV result, EAX
    push 0
    call ExitProcess
END START
```

2. Скомпилировать программу и получить исполняемый файл lab1.exe.
3. Открыть файл lab1.exe в отладчике.
4. Сегмент содержит строку ente и 5 переменных a, b, result, p и ten:



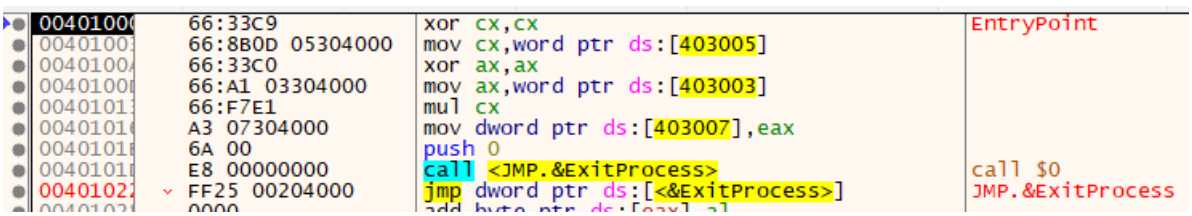
Адрес	Шестнадцатеричное				ASCII
00402000	0D 0A 00 05	00 05 00 00	00 00 00 00	00 00 00 00	……

00402010	00 00 00 00	00 00 00 00	F8 02 40 00	00 00 00 00	........ø.@.....
00402020	00 00 00 00 00	00 00 00 00 00	00 00 00 00 00	00 00 00 00 00	.....

Название перемен-ной	Начальный адрес	Конечный адрес	Размер данных, байт	Описание
ente	004020000	00402002	3	символы возврата каретки (13), перевода строки (10), окончания строки (0)
a	004020003	00402005	2	Объявляется как 16-битное слово (WORD) со значением 5.
b	00402006	00402006	2	Объявляется как 16-битное слово (WORD) со значением 5.
result	00402007	0040200A	4	Объявляется как 32-битное двойное слово (DWORD) без инициализации.
p	0040200B	00402010	6	Объявлена 6-байтовая целочисленная переменная
ten	00402011	0040201A	8	8-байтовая переменная с плавающей точкой
Общий размер сегмента данных:			**25**	

Ячейки памяти с адресами от 004020000 до 004020002 содержат ASCII символы возврата каретки (13), перевода строки (10) и окончания строки. Потом объявляются две переменные a и b на участке от 004020003 до 004020006 как слова (DW) 2 байта и каждая инициализирована значением 5. Дальше объявляется переменная result на участке от 00402007 до 0040200A как двойное слово (DD). Она еще не инициализирована, но в последующем коде используется для хранения результата умножения переменных a и b. Переменная p объявлена как 6-байтная целочисленная переменная (DF). Но так как она не может быть инициализирована как число с плавающей точкой, в памяти она представлена нулями. И последняя переменная ten определена как 8 байтовое число с плавающей точкой (DT)

5. Пошаговая трассировка программы



0. Исходное состояние регистров:

```
EAX   0019FFCC
EBX   002E9000
ECX   00401000     <lab1.EntryPoint>
EDX   00401000     <lab1.EntryPoint>
EBP   0019FF84
ESP   0019FF78
ESI   00401000     <lab1.EntryPoint>
EDI   00401000     <lab1.EntryPoint>

EIP   00401000     <lab1.EntryPoint>

EFLAGS   00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus 00000000 (STATUS_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B
```

1. Выполняет побитовую операцию «исключающее или» над числами в регистре CX: CX = CX xor CX. Обнуляет регистр CX. Увеличивает регистр EIP на 3:

xor cx, cx

```
EAX   0019FFCC
EBX   002E9000
ECX   00400000     lab1.00400000
EDX   00401000     <lab1.EntryPoint>
EBP   0019FF84
ESP   0019FF78
ESI   00401000     <lab1.EntryPoint>
EDI   00401000     <lab1.EntryPoint>

EIP   00401003     lab1.00401003

EFLAGS   00000246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus 00000000 (STATUS_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B
```

2. Выполнение этой команды приведет к тому, что двухбайтовое значение из памяти по адресу 403005 в сегменте данных будет скопировано в регистр cx. Увеличивает регистр EIP на 7:

mov cx,word ptr ds:[403005]

```
EAX   0019FFCC
EBX   002E9000
ECX   00400005     lab1.00400005
EDX   00401000     <lab1.EntryPoint>
EBP   0019FF84
ESP   0019FF78
ESI   00401000     <lab1.EntryPoint>
EDI   00401000     <lab1.EntryPoint>

EIP   0040100A     lab1.0040100A

EFLAGS   00000246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus 00000000 (STATUS_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B
```

3. Выполняет побитовую операцию «исключающее или» над числами в регистре AX: AX = AX xor AX. Обнуляет регистр AX. Увеличивает регистр EIP на 3:

xor ax, ax

```
EAX   00190000
EBX   002E9000
ECX   00400005     lab1.00400005
EDX   00401000     <lab1.EntryPoint>
EBP   0019FF84
ESP   0019FF78
ESI   00401000     <lab1.EntryPoint>
EDI   00401000     <lab1.EntryPoint>

EIP   0040100D     lab1.0040100D

EFLAGS   00000246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus 00000000 (STATUS_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B
```

2. Выполнение этой команды приведет к тому, что двухбайтовое значение из памяти по адресу 403003 в сегменте данных будет скопировано в регистр AX. Увеличивает регистр EIP на 6:

mov ax, word ptr ds:[0x00403003]

```
EAX   00190005
EBX   002E9000
ECX   00400005     lab1.00400005
EDX   00401000     <lab1.EntryPoint>
EBP   0019FF84
ESP   0019FF78
ESI   00401000     <lab1.EntryPoint>
EDI   00401000     <lab1.EntryPoint>

EIP   00401013     lab1.00401013

EFLAGS   00000246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus 00000000 (STATUS_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B
```

3. Команда mul cx выполняет умножение 16-битного числа в регистре cx на 16-битное число в регистре ax

mul cx

```
EAX   00190019
EBX   002E9000
ECX   00400005     lab1.00400005
EDX   00400000     lab1.00400000
EBP   0019FF84
ESP   0019FF78
ESI   00401000     <lab1.EntryPoint>
EDI   00401000     <lab1.EntryPoint>

EIP   00401016     lab1.00401016

EFLAGS   00000202
ZF 0  PF 0  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus 00000000 (STATUS_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B
```

4. Команда копирует 32-битное значение из регистра EAX в место памяти, на которое указывает адрес 0x00403007

mov dword ptr ds:[0x00403007], eax