

## Лабораторная работа asm3 (2 часа)

### Конструирование программного обеспечения

#### Создание программы на языке ассемблера

**Цель работы:** создание MASM-проекта в Visual Studio на языке ассемблера, с использованием косвенной адресации, команд переходов и циклов.

#### Задание.

1. Используйте материал лекции asm01 – asm04.
2. Создайте проект-приложение с именем **SE\_Asm03** на языке ассемблера.
3. Разработайте программу **asm03**:

- а) используйте следующие операторы определения данных:

```
myBytes    BYTE    10h, 20h, 30h, 40h
myWords    WORD    8Ah, 3Bh, 44h, 5Fh, 99h
myDoubles  DWORD    1, 2, 3, 4, 5, 6
myPointer   DWORD    myDoubles
```

- б) определите значения регистров после выполнения команд, в соответствии со своим вариантом:

1	<pre>mov ESI, OFFSET myBytes mov AH, [ESI] mov AL, [ESI + 3]</pre>	<pre>; AH = ? ; AL = ?</pre>
2	<pre>mov ESI, OFFSET myWords mov AX, [ESI + 4] mov BX, [ESI + 2]</pre>	<pre>; AX = ? ; BX = ?</pre>
3	<pre>mov ESI, 0 mov EAX, myDoubles[ESI] mov EDX, [myDoubles + ESI]</pre>	<pre>; EAX = ? ; EDX = ?</pre>
4	<pre>mov EDI, 2 mov AX, [myWords + EDI] mov BX, myWords[0]</pre>	<pre>; AX = ? ; BX = ?</pre>
5	<pre>mov EBX, myPointer mov EAX, [EBX + 4] mov EDX, [EBX]</pre>	<pre>; EAX = ? ; EDX = ?</pre>
6	<pre>mov ESI, OFFSET myBytes mov AH, [ESI] mov AL, [ESI + 2]</pre>	<pre>; AH = ? ; AL = ?</pre>

7	<code>mov ESI, OFFSET myWords mov AX, [ESI + 0] mov BX, [ESI + 2]</code>	<code>; AX = ? ; BX = ?</code>
8	<code>mov ESI, 0 mov EAX, myDoubles[ESI + 4] mov EDX, [myDoubles + ESI]</code>	<code>; EAX = ? ; EDX = ?</code>
9	<code>mov EDI, 2 mov AX, [myWords + EDI] mov BX, myWords[0]</code>	<code>; AX = ? ; BX = ?</code>
10	<code>mov EBX, myPointer mov EAX, [EBX + 8] mov EDX, [EBX]</code>	<code>; EAX = ? ; EDX = ?</code>
11	<code>mov ESI, OFFSET myBytes mov AH, [ESI + 1] mov AL, [ESI + 3]</code>	<code>; AH = ? ; AL = ?</code>
12	<code>mov ESI, OFFSET myWords mov AX, [ESI + 4] mov BX, [ESI + 2]</code>	<code>; AX = ? ; BX = ?</code>
13	<code>mov ESI, 0 mov EAX, myDoubles[ESI + 1] mov EDX, [myDoubles + ESI]</code>	<code>; EAX = ? ; EDX = ?</code>
14	<code>mov EDI, 4 mov AX, [myWords + EDI] mov BX, myWords[0]</code>	<code>; AX = ? ; BXS = ?</code>
15	<code>mov EBX, myPointer mov EAX, [EBX + 4] mov EDX, [EBX + 8]</code>	<code>; EAX = ? ; EDX = ?</code>

с) Дан массив любых целых чисел, размер массива равен 7.

- Вычислить сумму элементов массива. Результат запомнить в регистре EAX.
- Проверить наличие в массиве нулевого элемента. Если в массиве есть нулевой элемент, в регистр EBX поместить значение 0, в противном случае – значение 1.

4. Выполните программу **asm03** в режиме отладки.

5. Исследуйте выполнение программы с помощью окон отладчика: окно регистров, окно памяти, окно контрольные значения.

6. Дизассемблируйте код программы и исследуйте его.

### **Дополнительное задание.**

Часть «с» дополнить выводом в консоль окна с сообщением и кнопкой выхода. Заголовок окна: ФИО студента, курс, номер группы.

Текст сообщения соответствует результату проверки на наличие нулевого элемента в нем.

7. Создайте отдельный директорию с именем **SEAsm03** и скопируйте в него исходный файл проекта
8. В **Командной строке разработчика** ассемблируйте asm-файл проекта с помощью команды **ml**.
9. В **Командной строке разработчика** на основе результатов компиляции скомпонуйте exe-файл с помощью команды **link**.
10. Выполните exe-файл в командной строке разработчика.