

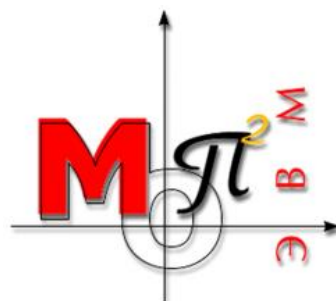
МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждения высшего  
образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ



## ЛАБОРАТОРНАЯ РАБОТА № 1

по дисциплине

«Машинно-ориентированное программирование»

на тему:

«Линейные программы»

Вариант № 9

Выполнил:  
Студент группы  
КТб02-8

Нестеренко П. А.

\_\_\_\_\_ *подпись*

Проверил:  
ассистент кафедры  
МОП ЭВМ

Гуляев Н. А.

\_\_\_\_\_ *подпись*

Оценка

« \_\_\_\_\_ » \_\_\_\_\_ 2020 г.

Таганрог 2020

# 1 ЦЕЛЬ РАБОТЫ

## 1.1 Дидактическая цель работы

Ознакомление с основными методами составления линейных программ на языке «Ассемблер», а также с наиболее часто используемыми мнемониками и регистрами процессора.

## 1.2 Практическая цель работы

В рамках лабораторной работы необходимо разработать программу на языке ассемблера, алгоритм которой выполняет задачу согласно описанному индивидуальному заданию. Скомпилировать и запустить программу с помощью пакета «**Tasm**» в DosBox

# 2 ВАРИАНТ ЗАДАНИЯ

## 2.1 Общие требования

Для всех вариантов требуется выполнить разработку программного модуля при помощи СРПО «Turbo Assembler», реализующего вычисление некоторого линейного уравнения, без вывода результата. Выполнить отладку разработанного программного модуля при помощи программного отладчика «**Turbo Debugger**».

## 2.2 Индивидуальное задание, вариант № 8

Разработать программу, реализующую вычисление неизвестного значения X в формуле:  $X = -7(C-A/4) + 3B - 5$ , исполнить программу с несколькими наборами исходных данных, проверить правильность результатов.

# 3 ХОД РАБОТЫ

## 3.1 Описание высокоуровневой реализации

В ходе работы была составлена программа на языке программирования «Ассемблер», использующая модель памяти «small», в которой допускается наличие одного сегмента кода и единственного сегмента данных. Размер стека программы – **256 байт**, что отражено в коде в виде строки «**stack 100h**».

Перед началом основного сегмента кода располагается инициализация констант «В», «С» и «Х». В первых двух константах пользователь может указать любые значения с максимальным размером в 2 байта (используется директива «dw»). Так как нам нужно найти «Х», в блоке инициализации необходимо присвоить ей «?» - знак вопроса, обозначающий отсутствие значения.

Логика основного сегмента кода заключается в том, что линейно происходит взаимодействие с двумя процессорными регистрами «АХ» и «ВХ». Сам процесс вычислений разбит на простые шаги с командами, которые процессор может обработать:

**mov X, Y** – пересылка из сегмента памяти X в Y.

**sal X, N** – арифметический сдвиг значения в сегменте памяти X влево на N единиц, с сохранением результата в X.

**add X, Y** – сложение значений из сегментов памяти X и Y с сохранением результата в X.

**sub X, Y** – вычитание значения из сегмента памяти Y из X с сохранением результата в X.

**sar X, N** – арифметический сдвиг значения в сегменте памяти X влево на N единиц, с сохранением результата в X.

После завершения работы с регистрами окончательный результат переносится из регистра «BX» в переменную «X». Полный листинг созданной программы расположен в «Приложение А». Результат вычислений можно увидеть с помощью программы «Turbo Debugger», занеся переменную «X» в список «Watches». В конце работы программа возвращает код успешного завершения работы – 0.

### 3.4 Описание полученных результатов

Программный модуль был дважды скомпилирован, запущен и отлажен в среде «DOS BOX» с разными входными данными. При вызове «Tasm» были заданы ключи «-L -ZI», которые позволили получить отладочные файлы. При вызове «TLINK» были использованы ключи «-V». После чего был запущен отладчик «TD», в котором было произведено пошаговое исполнение программ. На *рисунках 1 и 2* представлены результаты выполнения программ в отладчике – переменная, которую необходимо вычислить, расположена в секции «Watches». Обе программы, в результате тестирования, выдали верный результат.

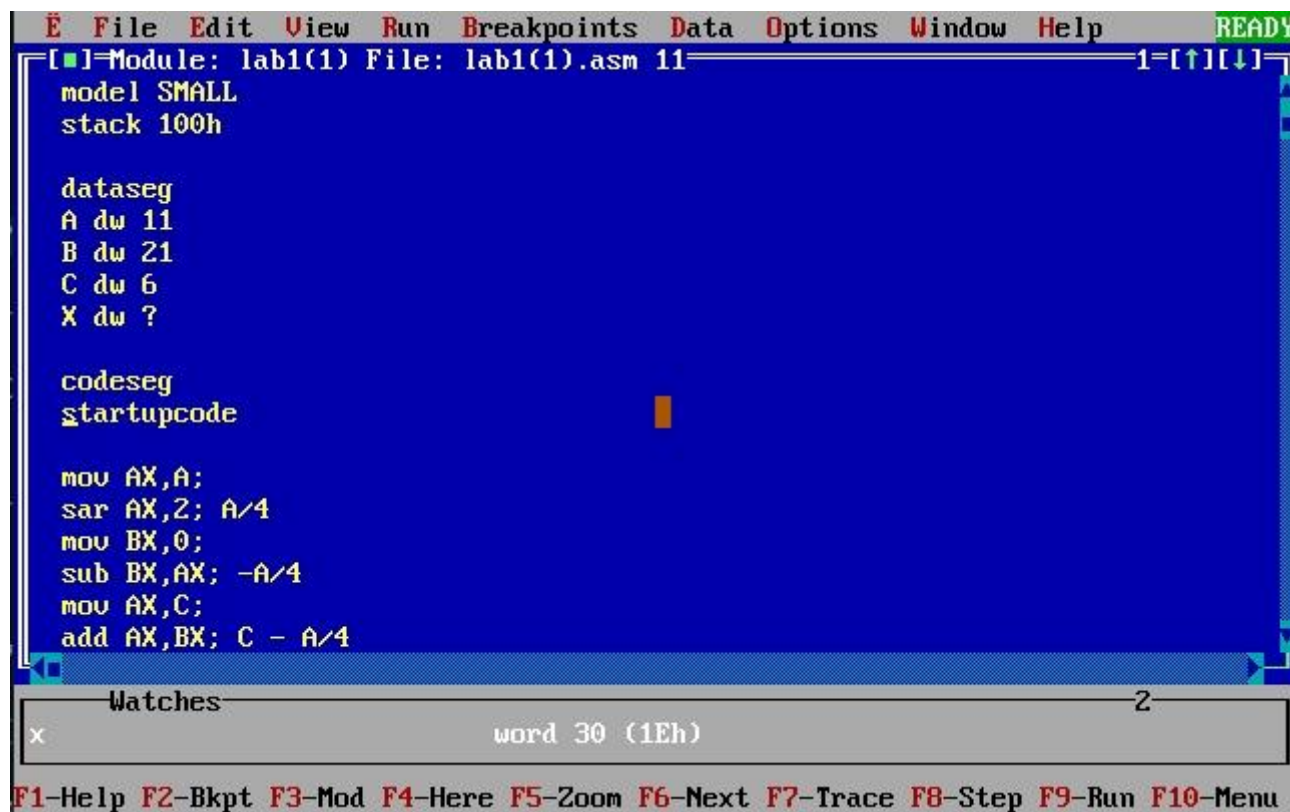
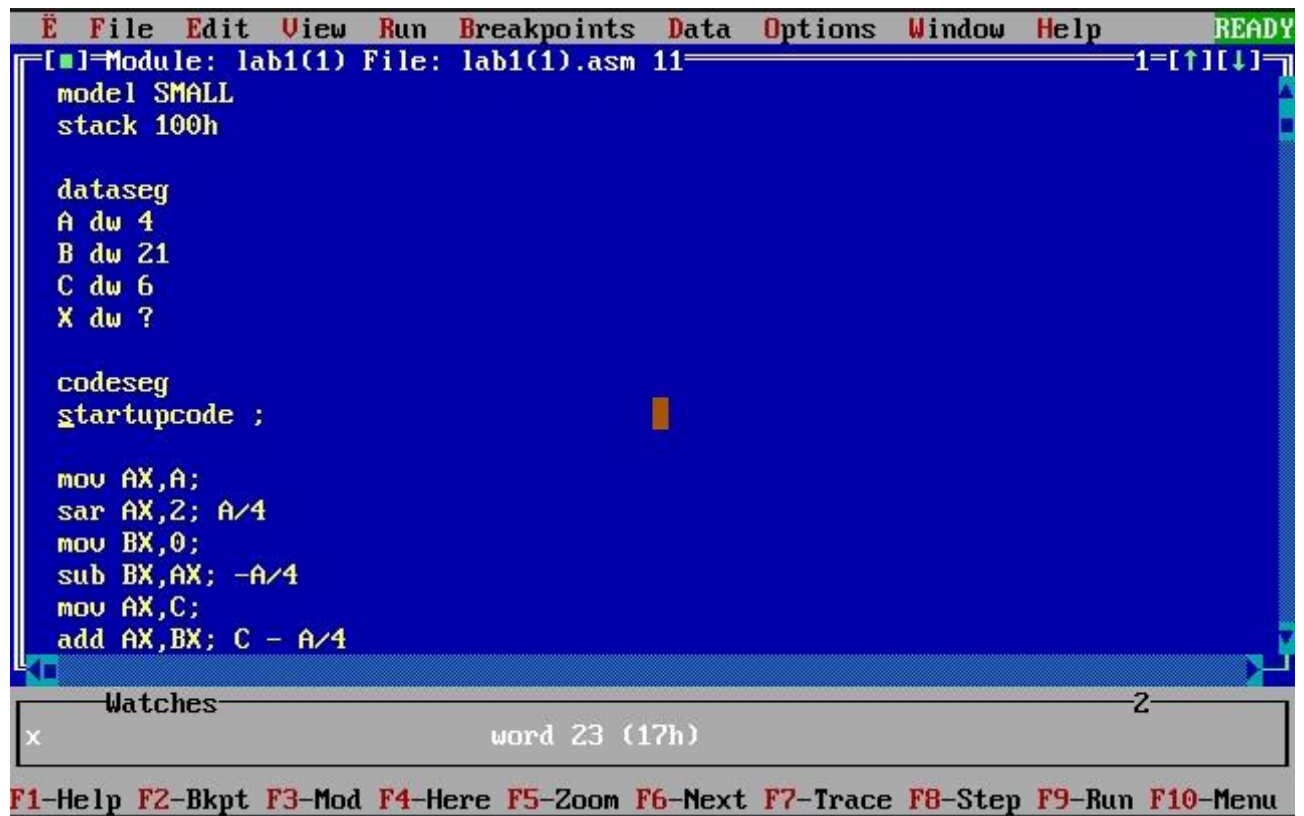


Рисунок 1 – Процесс отладки при помощи средства «Turbo Debugger» с первым набором входных данных



*Рисунок 2 – Процесс отладки при помощи средства «Turbo Debugger» с вторым набором входных данных*

## 4 ВЫВОДЫ

### 4.1 Полученные знания, навыки, умения

В ходе выполнения лабораторной работы был разработан и отлажен программный модуль, выполняющий вычисление заданного линейного уравнения с одним неизвестным.

## ПРИЛОЖЕНИЕ А

```
model SMALL
stack 100h
```

```
dataseg
A dw 11
B dw 21
C dw 6
X dw ?
```

```
codeseg
startupcode
```

```
mov AX,A;
sar AX,2; A/4
mov BX,0;
sub BX,AX; -A/4
mov AX,C;
add AX,BX; C - A/4
mov BX,AX;
sal AX,3;
sub BX,AX; -7(C-A/4)
mov AX,B;
add AX,AX;
add AX,B;
add BX,AX; -7(C - A/4) + 3B
sub BX,5;
mov X,BX; -7(C - A/4) + 3B - 5
```

```
QUIT: exitcode 0
end
```