

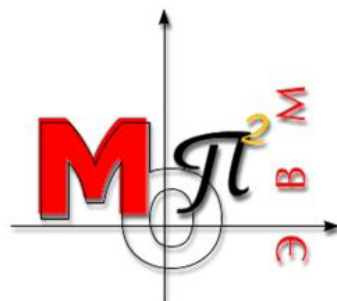
МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждения высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ



ЛАБОРАТОРНАЯ РАБОТА № 3

по дисциплине

«Машинно-ориентированное программирование»

на тему:

«Использование логических команд»

Вариант № 1

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 программ, используемых циклические алгоритмы работы, на языке «Ассемблер», а также с наиболее часто используемыми для этого мнемониками.

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 код программы с помощью программного пакета «TASM».

2 ВАРИАНТ ЗАДАНИЯ

2.1 Общие требования

Для всех вариантов требуется выполнить разработку программного модуля при помощи СРПО «Turbo Assembler», реализующего некоторую обработку массива, состоящего из байтов или 2-байтных слов. Выполнить отладку разработанного программного модуля при помощи программного отладчика «Turbo Debugger».

2.2 Индивидуальное задание, вариант № 3

Дан массив из 10 байтов. Посчитать количество байтов, в которых сброшены 6 и 4 биты.

3 ХОД РАБОТЫ

3.1 Описание высокоуровневой реализации

В ход работы была составлена программа на языке программирования «Ассемблер», используя модель памяти «small», в котором допускается наличие одного сегмента кода и единственного сегмента данных. Размер стека программы – 256 байт.

Перед началом основного сегмента кода происходит инициализация:

1. Переменной «COUNT», которая хранит в себе количество найденных по ходу работы программы единиц.
2. Массива «MB», представляющий из себя массив из 10 байт.

Алгоритм работы программы:

- 1) В регистр **CX**, который является счетчиком для цикла **loop** помещается число 10, что является длиной обрабатываемого массива:
mov CX 10.
- 2) В регистр **BX** помещается указатель на первый элемент массива:
lea BX MB.
- 3) Устанавливаем метку **BEG** для работы цикла.
- 4) Производим операцию **test AL,0101000B.**
- 5) Формируется флаг **jnz** в случае если биты 4 и 6 установлены и программа переключается на метку **NO**. Иначе значение регистра **DX** инкрементируется.
- 6) Эти действия повторяются в цикле 10 раз.

- 7) После окончания исполнения цикла командой **mov count, DX** мы помещаем искомое значение в созданную для этого переменную. Программа завершает своё выполнение.

Полный листинг программы расположен в «Приложение А».

3.4 Описание полученных результатов

Программный модуль был скомпилирован, запущен и отлажен в рамках среды «DOS BOX». При вызове «TASM» были заданы ключи «-L -ZI», которые позволили получить отладочные файлы. При вызове «TLINK» были использованы ключи «-V». С целью отладки был запущен отладчик «TD», в котором было проведено пошаговое исполнение программы. На рисунках 1 и 2 можно наблюдать результат выполнения программы в отладчике – значение переменной, которую необходимо вычислить расположено в секции «Watches». Результат работы программы совпал с ожидаемым – программа отработала корректно.

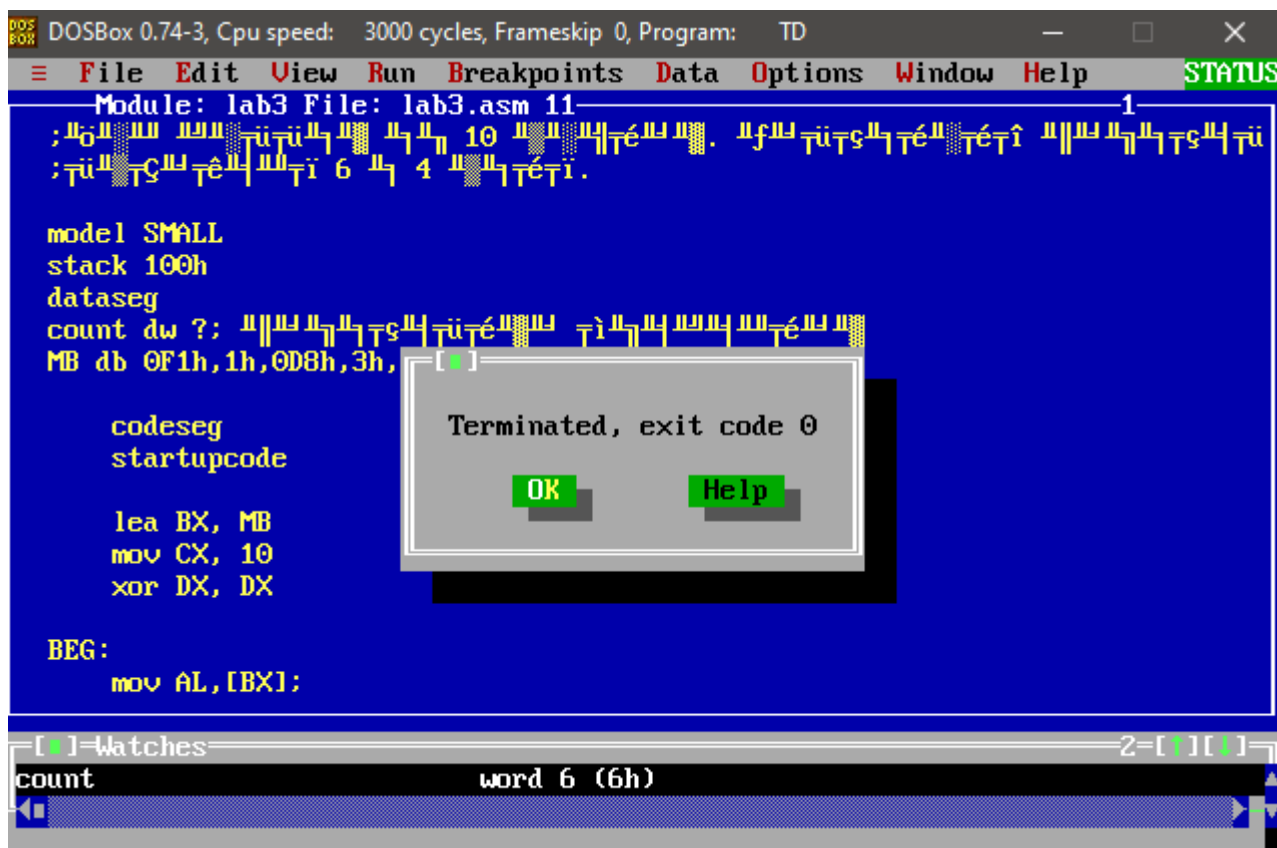


Рисунок 1 – результат работы программы в «Turbo Debugger»

4 ВЫВОДЫ

4.1 Полученные знания, навыки, умения

В ходе выполнения лабораторной работы была разработан и отлажен программный модуль, который считает количество байт, в которых

установлены биты 4 и 6. Получены навыки работы с основным логическими командами в программах на языке «Ассемблер».

ПРИЛОЖЕНИЕ А

;Дан массив из 10 байтов. Посчитать количество байтов, в которых
;сброшены 6 и 4 бита.

```
model SMALL
stack 100h
dataseg
count dw ?; количество элементов
MB db 0F1h,1h,0D8h,3h,50h,5h,55h,7h,0h,2Ah
```

```
codeseg
startupcode
```

```
lea BX, MB
mov CX, 10
xor DX, DX
```

```
BEG:
    mov AL,[BX];
    test AL, 01010000b
    jnz NO ; биты 4 и 6 установлены
    inc DX ; иначе увеличиваем счётчик
```

```
NO:
    inc BX;
    loop BEG;
; Конец цикла
```

```
mov count,DX;
```

```
QUIT: exitcode 0
end
```