

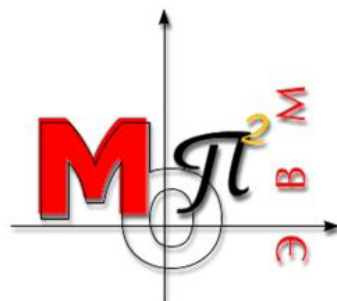
МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждения высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ



## ЛАБОРАТОРНАЯ РАБОТА № 4

по дисциплине

«Машинно-ориентированное программирование»

на тему:

«Обработка символьной информации»

Вариант № 4

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 методами составления программ на языке «Ассемблер», использующих ввод/вывод информации в консоли пользователя, а также с методами обработки этой информации.

## **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 символов, введенных из консоли, а так же вывод результата работы.

## **2.2 Индивидуальное задание, вариант № 3**

Ввести с клавиатуры строку, состоящую из целых чисел, разделенных пробелами. Найти количество чисел больших, чем первое число строки.

# **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 структур данных, необходимых для работы программы:

```

Ask db 0Ah,0Dh,'Vvedite stroku:', '$' ; Приглашение ввода
Answer db 0Ah,0Dh, 'Resultat:', '$' ; Сообщение для вывода
Counter db 0 ; Счетчик чисел меньше первого
Num db 0 ; переменная общего назначения)
Inpstr db 80, ?, 82 dup( ? ) ; буфер ввода
Outputstr db 0Ah, 0Dh, ?, ?, 0Dh, 0Ah, '$' ; буфер вывода
max db -100 ; число для хранения первой переменной массива

```

### Алгоритм работы программы:

```

codeseg
startupcode
Begin:
    lea DX, Ask
    mov AH, 09h
    int 21h
    lea DX, Inpstr
    mov AH, 0Ah
    int 21h
    ;Обработка
    lea BX, Inpstr+2 ;адрес начала введенной строки
    mov CL, Inpstr+1 ;кол-во введенных символов строки
    jmp Cycle0

Mult:
    ; Формируем число поразрядно
    mov AL, 10;
    mul Num
    mov Num, AL
    jmp Cycle1

Cycle0:
    mov AL, [BX];
    add BL, 1
    cmp AL, ' '
    je NC

    ;Если всё таки цифра

```

```
sub AL, '0'  
push AX  
jmp Mult
```

Cycle1:

```
pop AX  
add Num, AL;
```

loop Cycle0

NC:

```
cmp Num,0  
je NC1
```

```
; Если это первое число в массиве то устанавливаем  
cmp max,-100  
;cmp CL, Inpstr+1 ; это работает ТОЛЬКО для однозначных чисел  
jnz SRAV  
; записываем первое число массива в max  
mov AL, Num  
mov max, AL  
jmp Clear
```

SRAV:

```
; Сравниваем текущее число с первым  
mov AL,max
```

```
cmp Num,AL;  
js Clear;результат <0  
cmp Num,AL;  
jz Clear;результат равен 0  
inc Counter;
```

Clear:

```
mov Num, 0
```

NC1:

```
; Если символы в строке кончились, то печатаем результат  
; иначе продолжаем цикл  
cmp CL, 0
```

```

        je Print
        loop Cycle0

Print:
        lea DX, Answer
        mov AH, 09h
        int 21h

        xor AX, AX
        mov AL, Counter
        mov BL, 10
        div BL
        add AH, '0'
        add AL, '0'
        cmp AL, 0
        mov Outputstr+2, AL
        mov Outputstr+3, AH

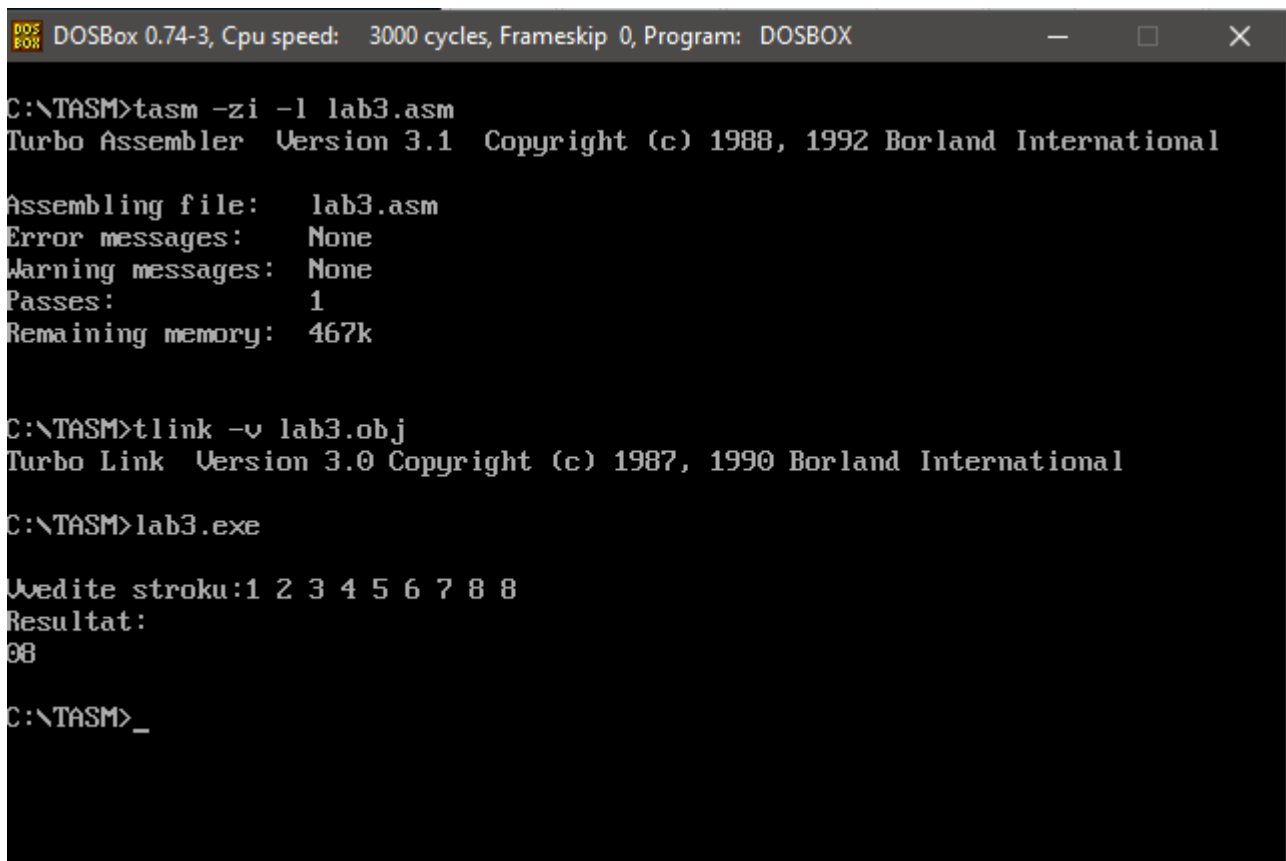
        lea DX, Outputstr
        mov AH, 09h
        int 21h
; В массив вывода переносим два байта с первой и второй цифрой счетчика
; Вызываем прерывание 21h для вывода в консоль буфера вывода

QUIT:
        exitcode 0
end

```

### 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. Результат работы программы совпал с ожидаемым – программа отработала корректно.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

C:\TASM>tasm -zi -l lab3.asm
Turbo Assembler Version 3.1 Copyright (c) 1988, 1992 Borland International

Assembling file: lab3.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 467k

C:\TASM>tlink -v lab3.obj
Turbo Link Version 3.0 Copyright (c) 1987, 1990 Borland International

C:\TASM>lab3.exe

Vvedite stroku:1 2 3 4 5 6 7 8 8
Resultat:
08

C:\TASM>_
```

*Рисунок 1 – результат работы программы*

## 4 ВЫВОДЫ

### 4.1 Полученные знания, навыки, умения

В ходе выполнения лабораторной работы была разработан и отлажен программный модуль, который считает некоторое количество чисел из консоли, определяет сколько из них больше первого и выводит результат работы в консоль.