**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10**

**студента гр. И-2-18**

Мартыненко Романа Сергеевича

По дисциплине: Программирование встроенных систем

Тема занятия: Поиск информации в оперативной памяти

Цель занятия: применение алгоритмов поиска информации в различных сегментах оперативной памяти

1. **Задание**

Адрес среды окружения находится в PSP со смещением 2Ch. Сегментный адрес PSP находится в регистрах DS и ES при старте программы.

Необходимо получить из среды окружения строки, содержащие заданный символ (символ задать программно), и вывести их на экран. Если строки/строк с заданным символом не существует – выдать сообщение «Строк с заданным символом не существует».

1. **Листинг файла** Lab\_10.asm

.model small            ;Модель памяти Small

.286                    ;16 разрядный проц

.stack 100h             ;Директива .STACK описывает сегмент стека

.data                   ;Идентификатор сегмента данных

    Sr00 db 'Line number No','$'

    Sr01 db ' does not exist.',10,13,'$'

    c dw 10             ;Делитель 10

.code

start:

    mov ax,[ds:2ch]     ;sreda okrujeniya

    mov ds,ax

    mov di,0            ; nomer tekushey stroki

    mov si,0            ; smesheniy v segmente sredy okrujeniya

    mov bp,1            ;nomer iskomoy stroki

k4: ; sravneniye nomera iskomoy stroki s nomerom tekushey stroki

    inc di

    cmp di,bp

    je k3

k1:

    mov dl,[ds:si]

    inc si

    cmp dl,0

    jne k1

    mov dl,[ds:si]

    inc si

    cmp dl,0

    jne k4

    jmp k2

k3:     ;vyvod stroki s iskomym nomerom

    mov ah,2h

    int 21h

    mov dl,[ds:si]

    cmp dl,0

    je k5

    inc si

    jmp k3

k2:     ; net stroki

    mov ax, @data           ;Передаем @data в ax

    mov ds, ax              ;Передаем ax в ds

    mov dx, offset Sr00     ;В dx записываем значение переменной Message

    mov ah,9h               ;в ah передаем код -9h (Вывод на экран)

    int 21h                 ;Вызываем прерывание для выполнения команды

    xor si, si          ;Обнуляем si

    mov ax, bp          ;Записываем bp в ax

k6:                     ;Метка начала блока k6

    xor dx, dx          ;Обнуление регистра dx (быстрее чем mov dx, 0)

    div c               ;Делим содержимое ax на c

    add dl, '0'         ;Прибовляем к младшему байту dx '0' (Преобразовали число к символу)

                        ;'0' = 48 (48-57 это номера символов от 0-9 в таблице ascii)

    mov cx, dx

    push cx             ;Оправляем cx в стек

    inc si              ;увеличиваем si на 1

    cmp ax, 0           ;Сравниваем ax с 0

    jne k6              ;Если ax не 0, то выполняем (k1) деление до тех пор пока число не кончится

k7:

    pop cx              ;Забираем из стека 2 байта и помещаем в cx

    mov dl, cl          ;в dl помещаем cl (1 байт)

    mov ah,2h           ;Вызов посимвольной отрисовки

    int 21h             ;Вызываем прерывание для выполнения команды

    dec si              ;Уменьшаем si на 1

    cmp si, 0           ;Сравниваем ax с 0

    jne k7              ;Если si не 0, то выполняем (k7) Отрисовка числа пока оно не закончится в стеке

    mov dx, offset Sr01     ;В dx записываем значение переменной Message

    mov ah,9h               ;в ah передаем код -9h (Вывод на экран)

    int 21h                 ;Вызываем прерывание для выполнения команды

k5:     ;exit

    mov ah, 4ch         ;Передаем код завершения программы

    int 21h             ;Вызываем прерывание для выполнения команды

end start               ;Окончания блока start

1. **Результат выполнения лабораторной работы**

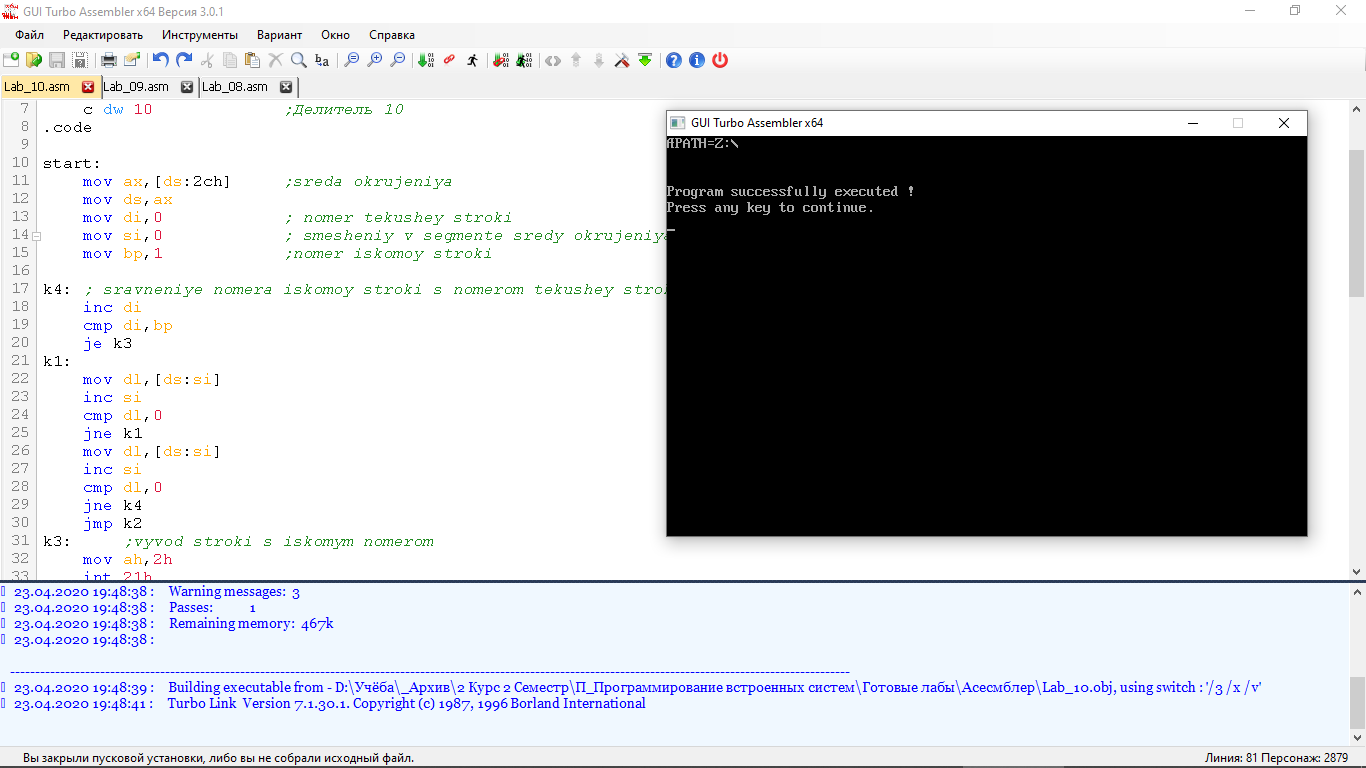


Рис 1. (Скриншот результата работы кода файла Lab\_10.asm) Строка найдена

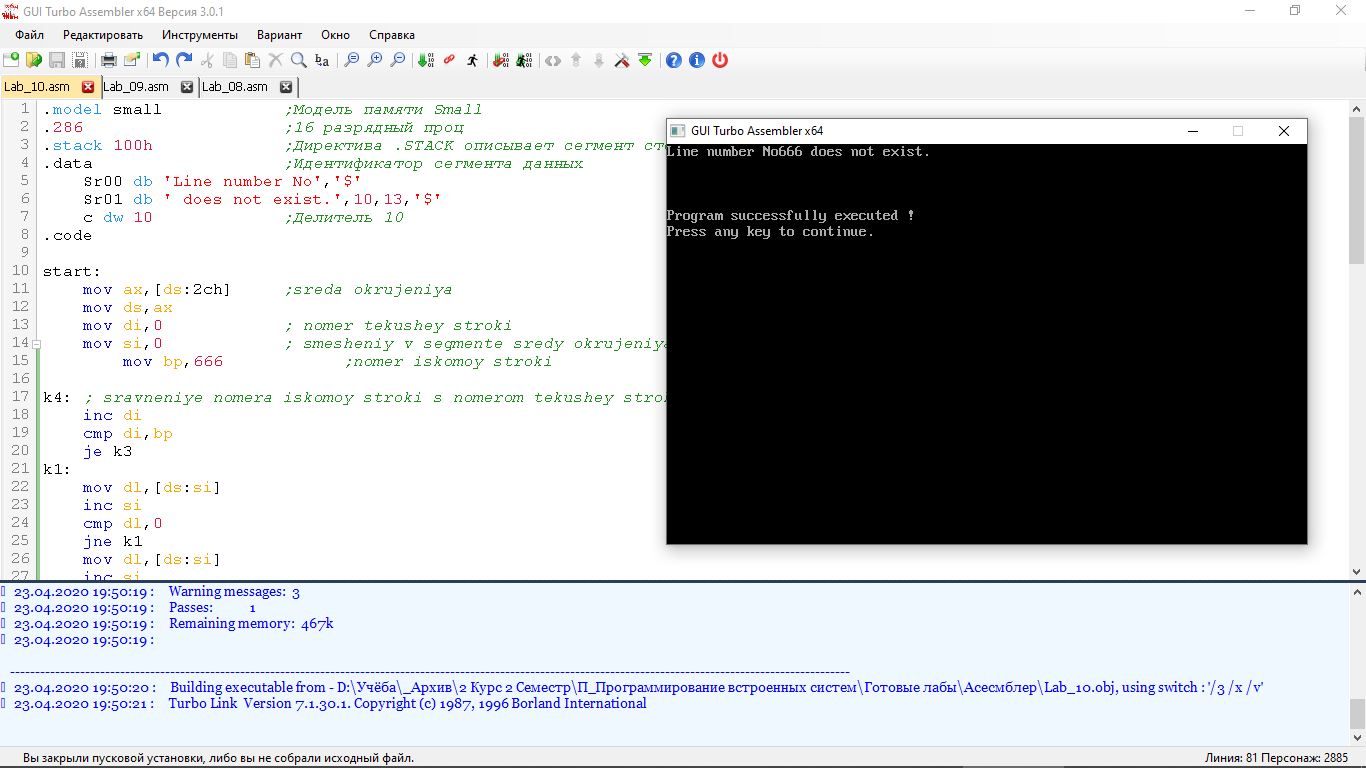


Рис 2. (Скриншот результата работы кода файла Lab\_10.asm) Строка не найдена