**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 9**

**Студентки гр. И-2-15(а)**

**Авдиль Салие**

**По дисциплине:** Разработка ПО для встроенных систем.

**Тема занятия:** Поиск информации в оперативной памяти.

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

**Задание:** Необходимо получить из среды окружения строку c заданным номером (номер строки задать программно). Если строки с заданным номером не существует – выдать сообщение «Строки с номером \_\_ не существует!».

**Листинг файла LAB9.asm**

**.model small**; *Указание модели памяти: указывает, что сегмент данных (он начинается с директивы.DATA) и сегмент кода (начинается с директивы.CODE) будут занимать не более одного сегмента ОЗУ ( 64 Кбайт).*

**.286**; *разрешены команды 286-ого микропроцессора*

**.data**; *директива описывает обычный сегмент данных (переменных).*

**a db 'Stroki s nomerom$'**

**b db 'ne sushestvuet!$'**

**c dw 9**

**d db 10,13,'$'**

**e db ' ', '$'** *; запись переменной в режиме строки*

**f dw 10**

**.code**; *объявление сегмента кода*

; *Метка входа в программу*

**start: mov ax, @data;** *пересылаем адрес сегмента данных в регистр ax,*

**mov ds, ax**; *устанавливаем регистр ds на сегмент данных*

**mov bp,c ;** *передаем содержимое переменной c в регистр bp*

**mov dx, offset d**; *смещение переменной d в dx*

**mov ah, 9h**

**int 21h ;** *2 перехода*

**int 21h**

**mov ax, [es:2ch];** *пересылаем в регистр ax адрес среды окружения* *со*

**mov ds, ax**; *смещением* 2ch *в* PSP

**mov di, 0**; *символ начала строки, обнуляем* di

**mov si, 0;** *текущий символ* , обнуляем si

**k1: inc di**; *увеличиваем di на единицу*

**cmp di, bp**; *сравниваем содержимое bp с di*

**je k3**; *переходим к k3 если равны*

**k2: mov dl, [ds:si]**; *пересылка в регистр dl символ среды окружения,*

**inc si**; *расположенный по смещение si, увеличиваем si на единицу*

**cmp dl, 0**; *если dl не равно переходим к k2*

**jne *k2***;

**mov dl, [ds:si]**; *пересылает в сегментный регистр ds адрес среды окружения, расположенный по смещению 2ch в PSP*

**cmp dl, 0**; *проверка на второй 0 в строке*

**jne *k1***; *если не равно переходим в k1*

**jmp *k4***; *иначе переходим в k4*

**k3: mov dl, 0**; *пересылаем в регистр dl 0*

**mov dl, [ds:si]**;*пересылает в сегментный регистр ds адрес среды окружения, расположенный по смещению 2ch в PSP.*

**mov al, 0**; *пересылаем в регистр al 0*

**cmp dl, 0**; *сравниваем содержимое dl c 0*

**je *k6***; *если равно то переходим в k6*

**mov ah, 2h**; *функция 02h(DOS)*

**int 21h**; *21-го прерывания для вывода символа*

**inc si** ; *увеличиваем на si единицу*

**jmp k3**; *переходим в k3*

**k4: mov ax, @data**;*пересылаем в регистр ax адрес сегмента данных*

**mov ds, ax**; *в регистр ds пересылаем содержимое ax*

**mov si, 4**; *заполняем строку*

**mov ax, bp**; *пересылаем содержимое bp в ax*

**mov dx, 0**; *обнуляем регистр dx*

**k5: div f**; *запуск цикла с ограничением, пока не 0, делим номер строки на 10 без остатка*

**add dl, 30h**; *прибавляем к dl 30h*

**mov [d+si], dl**; *и записываем в последний байт*

**dec si**; *уменьшаем si на единицу*

**mov dx, 0**; *пересылаем в регистр dx 0*

**cmp ax, 0**; *сравниваем содержимое ax с 0*

**jne *k5***; *если не равно переходим в k5*

**mov dx, offset a**; *смещение переменной a в регистр dx*

**mov ah, 9h**; *9-ая функция 21-го прерывания для вывода*

**int 21h**; *строки на экран*

**mov dx, offset e**; *смещение переменной e в регистр dx*

**mov ah, 9h**; *9-ая функция 21-го прерывания для вывода*

**int 21h**; *строки на экран*

**mov dx, offset b**; *смещение переменной b в регистр dx*

**mov ah, 9h**; *9-ая функция 21-го прерывания для вывода*

**int 21h**; *строки на экран*

**k6: mov ah, 4ch**; *Завершение программы*

**int 21h**

**end start**

**Результат выполнения файла LAB9.asm:**

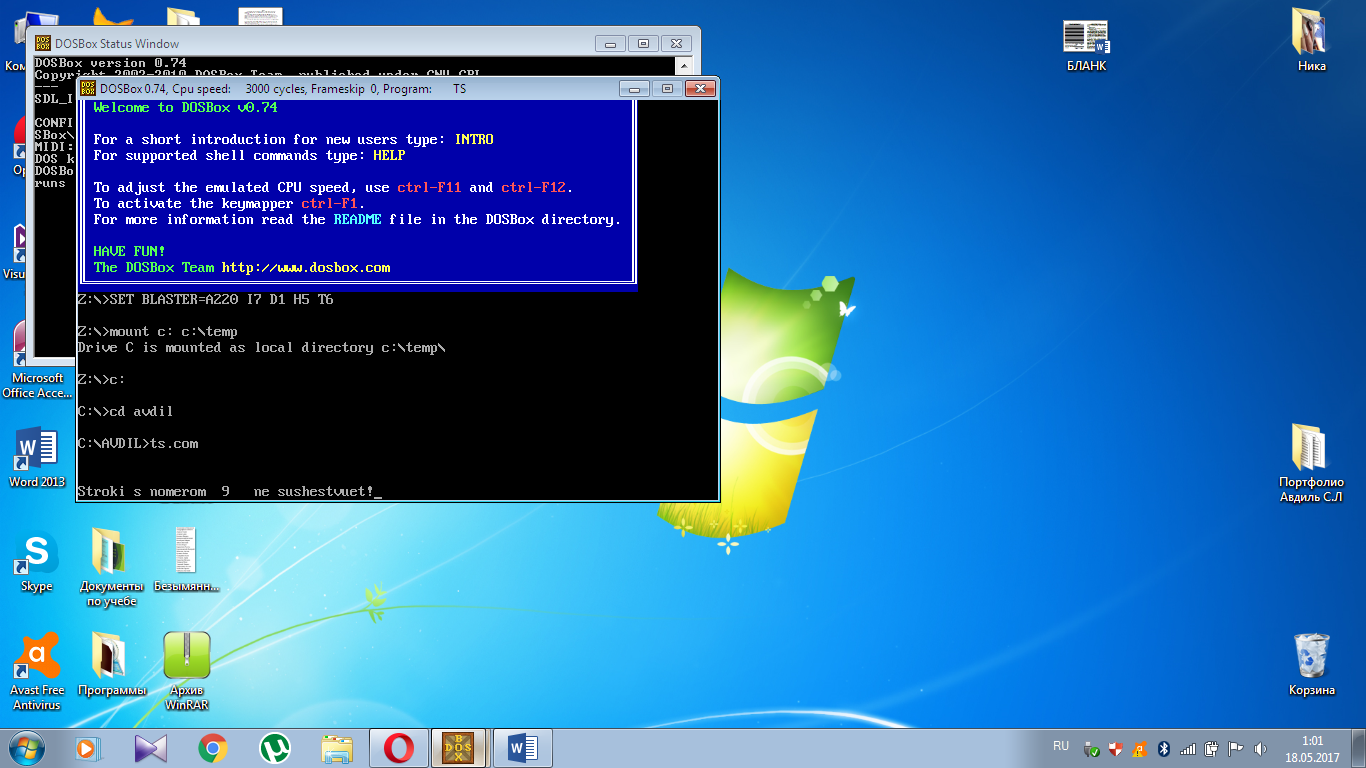


Рисунок 1. Вывод информации, что данной строки не существует