**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное автономное образовательное  
 учреждение высшего образования   
«Южный федеральный университет»**

**Институт высоких технологий и пьезотехники**

**Кафедра прикладной информатики и инноватики**

**Направление подготовки:   
09.03.03 "Прикладная информатика"**

**Отчет по Лабораторной работе №5**

**«Функции DOS»**

**По дисциплине «Основы функционирования вычислительной техники»**

**Выполнил:**

студент 2 курса 6 группы

Куракин Н. А.

\_\_\_\_\_\_\_\_\_\_\_\_

*подпись*

**Проверил:**

Преподаватель Толмачев С.А.

Ростов-на-Дону

2024

# **ЛАБОРАТОРНАЯ РАБОТА №5**

**Тема:** «Функции DOS».

**Цель работы:** изучение функций DOS, их применение в программах на

языке ассемблера.

**Ход работы:**

Ознакомился с теоретической частью и приступил к выполнению задания. Необходимо было разработать программу, реализующую **17 задание:**

**Ввести с клавиатуры строку и некоторое число. Если строка длиннее заданного числа, то обрезать, если короче растянуть, вставив нужное число пробелов между словами. Вывести результаты на экран.**

Внес в сегмент данных исходные данные

После этого в сегмент кода внес 2 ввода строк. Для строки и числа.

Т.к. число вносится десятичное, необходимо переделать его в 16-ричную систему счисления. Реализовал данную проблему

Затем сравнил длину строки и число, вводимое пользователем. Если число меньше или равно строке, то "копируется" количество символов, равное введенному числу.

Иначе (если число больше) считаются пропуски между словами.

Если этих пропусков 0, то в НАЧАЛО строки добавляются пробелы, а потом сама строка.

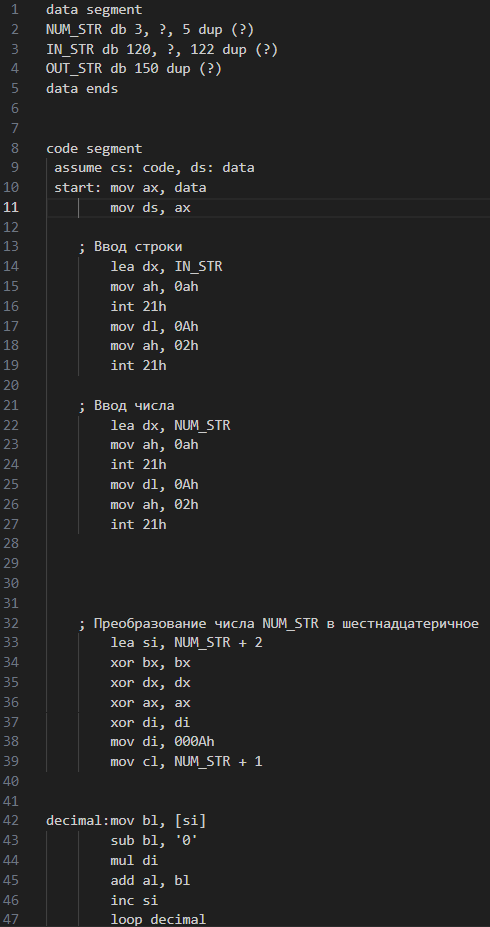
В ином случае из числа вычитается длина строки. Получившееся делится на "пропуски".

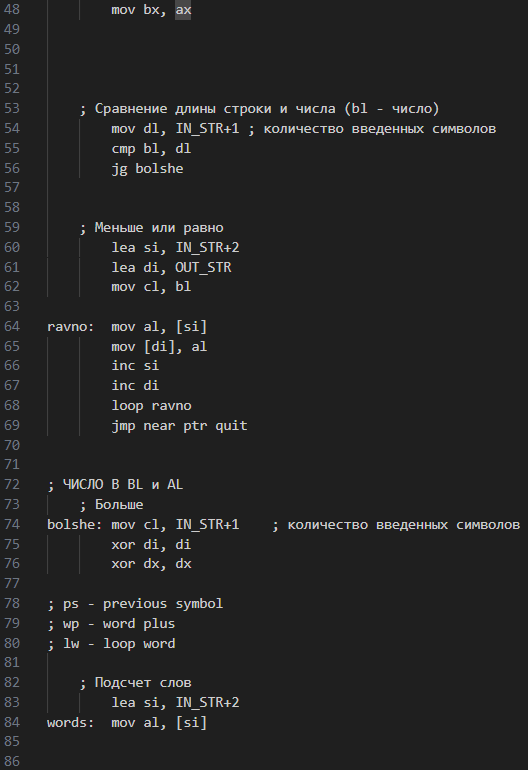
После этого посимвольно строится строка. Сначала добавляется символ, затем проверяется: является ли данный символ буквой или нет. Если да, переход к следующему символу.

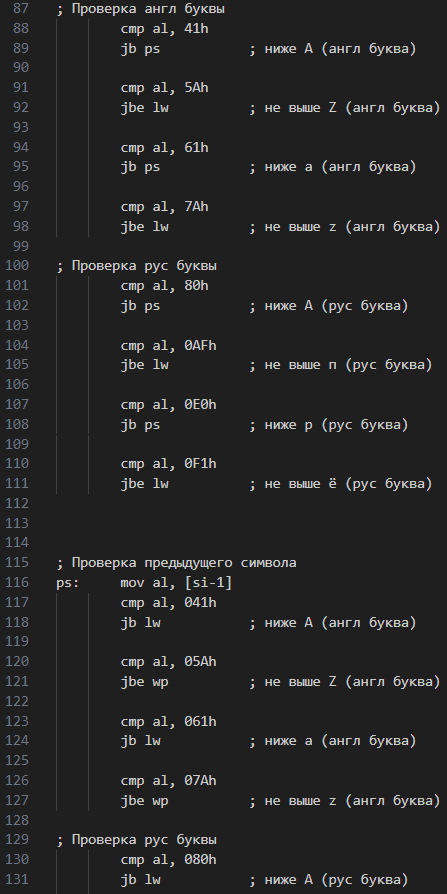
Если нет, то проверяется предыдущий символ: если он буква, то добавляется столько пробелов, сколько получилось в ax (al) после деления. Затем проверяется остаток: если он не равен 0, то добавить ещё 1 пробел. Перейти к следующему символу.

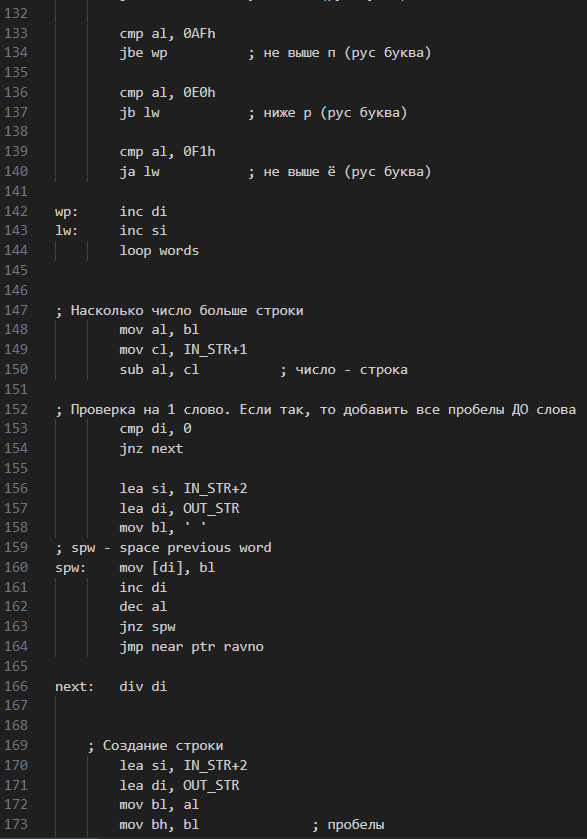
Когда строка закончится, то переход на последнюю метку quit. В ней находится вывод данной строки и выход из самой программы.

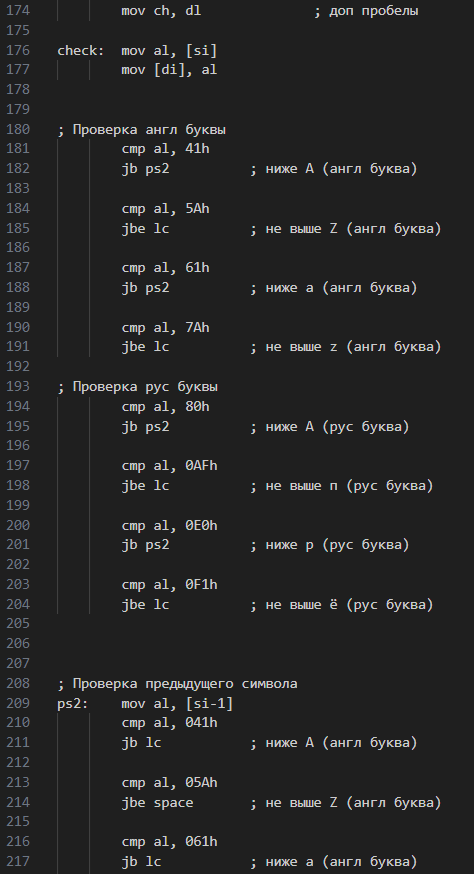
В результате полный код выглядит следующим образом (скрином):

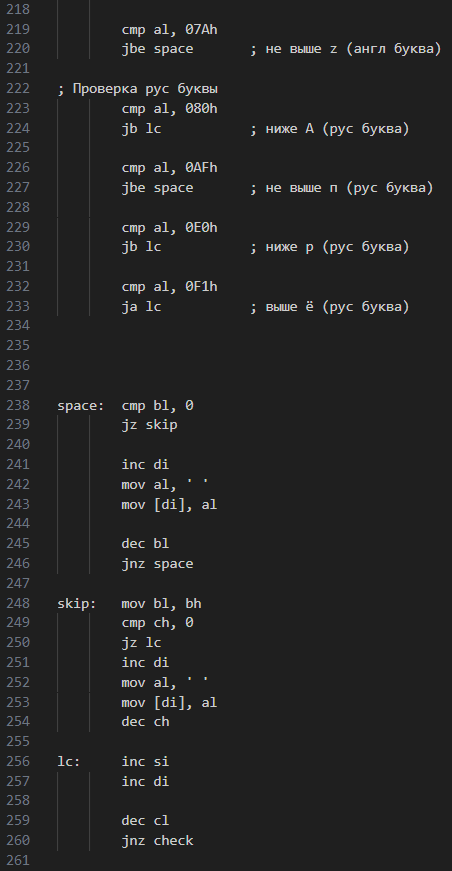


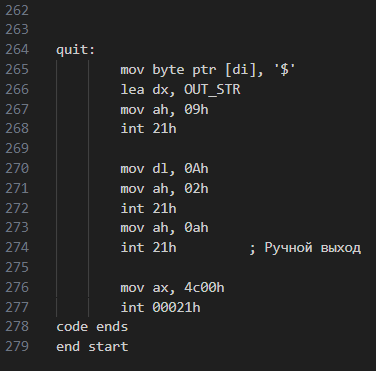












Этот же код текстом:

data segment

NUM\_STR db 3, ?, 5 dup (?)

IN\_STR db 120, ?, 122 dup (?)

OUT\_STR db 150 dup (?)

data ends

code segment

assume cs: code, ds: data

start: mov ax, data

mov ds, ax

; Ввод строки

lea dx, IN\_STR

mov ah, 0ah

int 21h

mov dl, 0Ah

mov ah, 02h

int 21h

; Ввод числа

lea dx, NUM\_STR

mov ah, 0ah

int 21h

mov dl, 0Ah

mov ah, 02h

int 21h

; Преобразование числа NUM\_STR в шестнадцатеричное

lea si, NUM\_STR + 2

xor bx, bx

xor dx, dx

xor ax, ax

xor di, di

mov di, 000Ah

mov cl, NUM\_STR + 1

decimal:mov bl, [si]

sub bl, '0'

mul di

add al, bl

inc si

loop decimal

mov bx, ax

; Сравнение длины строки и числа (bl - число)

mov dl, IN\_STR+1 ; количество введенных символов

cmp bl, dl

jg bolshe

; Меньше или равно

lea si, IN\_STR+2

lea di, OUT\_STR

mov cl, bl

ravno: mov al, [si]

mov [di], al

inc si

inc di

loop ravno

jmp near ptr quit

; ЧИСЛО В BL и AL

; Больше

bolshe: mov cl, IN\_STR+1 ; количество введенных символов

xor di, di

xor dx, dx

; ps - previous symbol

; wp - word plus

; lw - loop word

; Подсчет слов

lea si, IN\_STR+2

words: mov al, [si]

; Проверка англ буквы

cmp al, 41h

jb ps ; ниже A (англ буква)

cmp al, 5Ah

jbe lw ; не выше Z (англ буква)

cmp al, 61h

jb ps ; ниже a (англ буква)

cmp al, 7Ah

jbe lw ; не выше z (англ буква)

; Проверка рус буквы

cmp al, 80h

jb ps ; ниже A (рус буква)

cmp al, 0AFh

jbe lw ; не выше п (рус буква)

cmp al, 0E0h

jb ps ; ниже р (рус буква)

cmp al, 0F1h

jbe lw ; не выше ё (рус буква)

; Проверка предыдущего символа

ps: mov al, [si-1]

cmp al, 041h

jb lw ; ниже A (англ буква)

cmp al, 05Ah

jbe wp ; не выше Z (англ буква)

cmp al, 061h

jb lw ; ниже a (англ буква)

cmp al, 07Ah

jbe wp ; не выше z (англ буква)

; Проверка рус буквы

cmp al, 080h

jb lw ; ниже A (рус буква)

cmp al, 0AFh

jbe wp ; не выше п (рус буква)

cmp al, 0E0h

jb lw ; ниже р (рус буква)

cmp al, 0F1h

ja lw ; не выше ё (рус буква)

wp: inc di

lw: inc si

loop words

; Насколько число больше строки

mov al, bl

mov cl, IN\_STR+1

sub al, cl ; число - строка

; Проверка на 1 слово. Если так, то добавить все пробелы ДО слова

cmp di, 0

jnz next

lea si, IN\_STR+2

lea di, OUT\_STR

mov bl, ' '

; spw - space previous word

spw: mov [di], bl

inc di

dec al

jnz spw

jmp near ptr ravno

next: div di

; Создание строки

lea si, IN\_STR+2

lea di, OUT\_STR

mov bl, al

mov bh, bl ; пробелы

mov ch, dl ; доп пробелы

check: mov al, [si]

mov [di], al

; Проверка англ буквы

cmp al, 41h

jb ps2 ; ниже A (англ буква)

cmp al, 5Ah

jbe lc ; не выше Z (англ буква)

cmp al, 61h

jb ps2 ; ниже a (англ буква)

cmp al, 7Ah

jbe lc ; не выше z (англ буква)

; Проверка рус буквы

cmp al, 80h

jb ps2 ; ниже A (рус буква)

cmp al, 0AFh

jbe lc ; не выше п (рус буква)

cmp al, 0E0h

jb ps2 ; ниже р (рус буква)

cmp al, 0F1h

jbe lc ; не выше ё (рус буква)

; Проверка предыдущего символа

ps2: mov al, [si-1]

cmp al, 041h

jb lc ; ниже A (англ буква)

cmp al, 05Ah

jbe space ; не выше Z (англ буква)

cmp al, 061h

jb lc ; ниже a (англ буква)

cmp al, 07Ah

jbe space ; не выше z (англ буква)

; Проверка рус буквы

cmp al, 080h

jb lc ; ниже A (рус буква)

cmp al, 0AFh

jbe space ; не выше п (рус буква)

cmp al, 0E0h

jb lc ; ниже р (рус буква)

cmp al, 0F1h

ja lc ; выше ё (рус буква)

space: cmp bl, 0

jz skip

inc di

mov al, ' '

mov [di], al

dec bl

jnz space

skip: mov bl, bh

cmp ch, 0

jz lc

inc di

mov al, ' '

mov [di], al

dec ch

lc: inc si

inc di

dec cl

jnz check

quit:

mov byte ptr [di], '$'

lea dx, OUT\_STR

mov ah, 09h

int 21h

mov dl, 0Ah

mov ah, 02h

int 21h

mov ah, 0ah

int 21h ; Ручной выход

mov ax, 4c00h

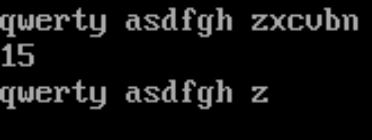
int 00021h

code ends

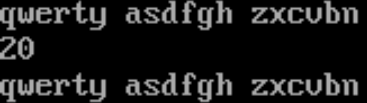
end start

Выполнил проверку на строке длиной в 20 символов (18 букв + 2 пробела):

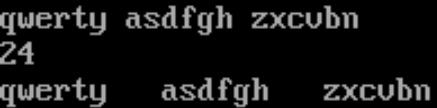
Строка > Число



Строка = Число



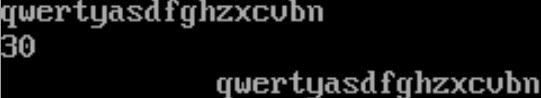
Строка < Число (нацело)



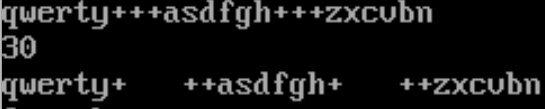
Строка < Число (с остатком)



Строка < Число (1 слово)



Строка < Число (3 слова + 6 пропусков, т.е. по 3 пропуска между словами)



\*Примечание: отображение с табуляцией может выглядеть некорректно. Связано с тем, что Tab занимает все места для символов до места кратного 8 (т.е. до 8, 16, 24, 32 и т.д. символа), однако, как символ строки и таблицы ASCII он считается лишь одним символом. Вследствие чего он может переместиться с 7 места (занимал 2 отображаемых символа, 7-8) на 9 (теперь занимает 8 символов, 9-16).