Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет

Лабораторная работа №2

Выполнил студент гр. 150505: Горбачевский К.В

Проверил: Туровец Н.О.

Минск 2022

Тема работы: Обработка символьных данных.

Цель работы: Ознакомиться с директивами определения данных, изучить команды пересылки данных и передачи управления, изучить строчные операции и прерывания консольного ввода-вывода высокого уровня.

Вариант лабораторной работы: Выполнить реверс символов слов строки.

Дополнительные требования к выполнению работы:

1. Выделить буфер для хранения 200 символов.

2. Строку символов ввести с клавиатуры, при этом ввод строки символов может быть завершен клавишей «Enter» или по заполнению буфера полностью.

3. Дополнительный буфер для хранения промежуточных результатов обработки строки в памяти не выделять.

4. При использовании констант задавать их с помощью директивы EQU.

5. Старт программы, ввод-вывод данных и обработку ошибок оформлять выводом в консоль поясняющих строк.

**Код программы**

data segment

str db 200,0, 200 dup ('$')

startProgram db '\_\_\_\_Start\_\_\_\_$'

enterString db 'Enter string: $'

revertString db 'Revert string: $'

lastStackElement equ 49

data ends

code segment

newLine:

mov ah, 02h

mov dl, 0ah

int 21h

mov ah, 02h

mov dl, 0dh

int 21h

ret

start: mov ax, data

mov ds, ax

mov ah, 9h

mov dx, offset startProgram

int 21h

call newLine

call newLine

mov ah, 9h

mov dx, offset enterString

int 21h

mov ah,0aH ; read string from keybord

mov dx, offset str ; into buffer

int 21h

mov di, offset str ; di = buffer address

mov bx, 0h ; clear bx register

mov bl, [di+1] ; bl = real string length

mov byte ptr [di+bx+2], '$' ; last symbol = '$'

call newLine

call newLine

mov ah, 9h

mov dx, offset revertString

int 21h

mov si, bx

add si, 2

mov cx,bx ; length of string

inc cx

push lastStackElement

stack: cmp str[si],' ' ; if symbol == ' '

je pushing

cmp str[si],'$' ; or symbol == '$'

je pushing

jne next

pushing: ; push index of last symbol of each word

push si

jmp next

next:

dec si

loop stack

word: pop bx ; pop index of last space before each word

mov dl, ' '

int 21h

cmp bx, lastStackElement

je endProgram

jne print

print:

mov cx, bx

add cx, -2 ; cycle for length - 2

add bx, -1 ; index of space - 1

printSymbol:

cmp str[bx], ' '

je word

jne nextSymbol

nextSymbol:

cmp str[bx], ' '

je word

mov ah, 02h

mov dl, str[bx]

int 21h

dec bx

loop printSymbol

jmp word

endProgram: int 10h

code ends

end start

**Вывод в консоли после компиляции**

