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: Лисичкин Д.А.

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:** 11.Заменить заданное слово в строке на другое заданное слово.

**Код программы**

data segment

enterFullString db " Enter your full string...$"

enterOldWord db " Enter old word...$"

enterNewWord db " Enter new word...$"

newString db " Your new string:$"

outofbuffer db " There is no such word in string!$"

otst db ' ',0Dh,0Ah,'$'

empty db " The string is empty!$"

buffer db 200,200 dup ('$')

tosearch db 200,200 dup('$')

toreplace db 200,200 dup('$')

ends

stack segment

db 100

ends

code segment

macro input buffer

push ax

push dx

mov [buffer], 200

mov [buffer + 1], 0

lea dx, buffer

mov ah, 0ah

int 21h

mov al, [buffer + 1]

add dx, ax

mov dx, '$'

pop dx

pop ax

endm

macro output string

lea dx, string

add dx, 2

mov ah, 9

int 21h

endm

macro endl

output otst

endm

macro replace 1str, 2str, wordtoadd

xor cx,cx

lea di, 1str+2

mov ax, di

lea si, 2str+2

mov ch, [2str+1]

cmp [di], 0dh

je string\_is\_empty

go:

mov dl, [di]

cmp dl, [si]

jne find\_new\_word

je continue

find\_new\_word:

cmp [di], '$'

je out\_of\_buffer

skip\_word:

inc di

cmp [di], ' '

jne find\_new\_word

skip\_spaces:

inc di

cmp [di], ' '

je find\_new\_word

lea si, 2str+2

mov ax, di

jmp go

continue:

inc di

inc si

cmp [si], 0Dh

je replace

jne go

replace:

check\_if\_not\_subword:

cmp [di], 20h

je continue2

cmp [di], 24h

je continue2

cmp [di], 9h

je continue2

cmp [di], 0Dh

je continue2

jmp find\_new\_word

continue2:

push 24h

push\_to\_stack:

lea bx, 1str

xor cx, cx

mov cl, [1str+1]

inc cl

add bx, cx

inc bx

pushing:

cmp bx, di

je continue3

mov dx, es:[bx]

push dx

mov es:[bx], '$'

dec bx

jmp pushing

continue3:

lea si, wordtoadd+2

mov di, ax

cld

mov cl, [wordtoadd+1]

repe movsb

mov dl, ' '

mov [di], dl

inc di

pop\_from\_stack:

pop dx

cmp dl, '$'

je end

mov es:[di], dl

inc di

jmp pop\_from\_stack

out\_of\_buffer:

output outofbuffer

endl

jmp end

string\_is\_empty:

output empty

endl

jmp end

end:

mov dl, '$'

mov es:[di], dl

endm

start:

mov ax, data

mov ds, ax

mov es, ax

output enterFullString

endl

input buffer

endl

output enterOldWord

endl

input tosearch

endl

output enterNewWord

endl

input toreplace

endl

replace buffer tosearch toreplace

output newString

endl

output buffer

endl

mov ah, 1

int 21h

mov ax, 4c00h

int 21h

ends

end start

**Вывод в консоли после компиляции**

