Министерство образования и науки Российской Федерации Федеральное агентство по образованию Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №1

по курсу «Системное программное обеспечение»

**Вариант 38**

Выполнил студент группы ИВТ-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М/

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Караваева О.В./

1. Задание:

Ввести строку символов. Сформировать новую строку, в которой символы разделены пробелами. Вывести обе строки на экран в режиме телетайпа с использованием атрибутов цвета.

2. Код

.model small

.data

message0 db 'String: $'

message1 db 0Ah, 0Dh, 'Symbol: $'

message2 db 0Ah, 0Dh, 'Result: $'

FullData STRUC

one db 11 dup(36) ; name type count init

two db 11 dup(36)

chr db 1 dup(36)

FullData ENDS

FD FullData 1 dup(<>)

.stack 256h

.code

GETSTRING proc near ; string will be written to [bx]

push cx

push dx

mov cx, 0Ah ; string length

mov dx, 01h ; break after Enter

getstring\_loop:

call GETCHAR ; get next char

cmp byte ptr [bx], 0Dh ; test if char is Enter

je getstring\_quit

inc bx

loop getstring\_loop

getstring\_quit:

pop dx

pop cx

ret

GETSTRING endp

GETCHAR proc near ; char will be written to [bx]

push ax

push dx

getchar\_begin:

mov ah, 08h

int 21h ; read char

cmp al, 00h ; test if char is functional

jne getchar\_obtain\_char

mov ah, 08h

int 21h

jmp getchar\_begin

getchar\_obtain\_char:

cmp dx, 01h

jne getchar\_test\_control

cmp al, 0Dh ; test if char is Enter

je getchar\_write\_char

getchar\_test\_control:

cmp al, 31 ; test if char is control

jle getchar\_begin

getchar\_write\_char:

mov [bx], al

mov dl, al ; write symbol to stdout

mov ah, 02h

int 21h

pop dx

pop ax

ret

GETCHAR endp

MESSAGE proc near ; data located in dx

push ax

push cx

mov ah, 09h ; write string to stdout

int 21h

pop cx

pop ax

ret

MESSAGE endp

main:

mov ax, @data

mov ds, ax ; load data segment

lea dx, message0

call MESSAGE

lea bx, FD.one

call GETSTRING

lea dx, message1

call MESSAGE

lea bx, FD.chr

xor dx, dx

call GETCHAR

mov al, FD.chr

xor dx, dx ; char counter

mov cx, 0Ah; loop counter

lea bx, FD.one

; count of char FD.chr in FD.one

main\_loop1\_begin:

cmp [bx], al

jne main\_loop1\_end

inc dx

main\_loop1\_end:

inc bx

loop main\_loop1\_begin

lea di, FD.one

cmp dx, 01h ; if count FD.chr is not greater then

jle main\_print\_string1 ; write original string to stdout

; copy dx chars to begin of the string

mov cx, dx

lea di, FD.two

main\_loop2\_begin:

mov [di], al

inc di

main\_loop2\_end:

loop main\_loop2\_begin

; copy other chars

mov cx, 0Ah

lea si, FD.one

lea di, FD.two

add di, dx

main\_loop3\_begin:

cmp byte ptr [si], al

je main\_loop3\_end

mov ah, [si]

mov byte ptr [di], ah

inc di

main\_loop3\_end:

inc si

loop main\_loop3\_begin

lea di, FD.two

main\_print\_string1:

lea dx, message2

call MESSAGE

mov byte ptr [di + 10], 24h

mov dx, di

call MESSAGE

mov ah, 4Ch

int 21h

end main