Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина «Конструирование программ»

**ОТЧЕТ**

к лабораторной работе №3

на тему:

**«СОЗДАНИЕ ПРОСТОЙ ПРОГРАММЫ НА ЯЗЫКЕ АССЕМБЛЕРА. ОБРАБОТКА СИМВОЛЬНЫХ ДАННЫХ.»**

БГУИР 6-05-0612-02 39

|  |
| --- |
| Выполнил студент группы 353503  ДЕРГУН Николай Олегович |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 3.** Написать программу для вывода на экран нескольких строк символов. Создать ассемблерные программы для двух вариантов исполняемых модулей (com и exe).

**Задание 2. Вариант 5.** Отсортировать слова в строке по значению ASCII кодов символов.

# 2 Выполнение работы

Ясно, что для настолько простой задачи больше подходит com формат файла, который не хранит дополнительной информации о программе и может использовать только один сегмент памяти, но условие требует создание программ для двух типов исполняемых модулей.

Листинг кода 1 – Текст программы задания 1 формата com

.model tiny

.code

main proc

org 100h

mov ah, 09h

mov dx, offset str1

int 21h

call nl

mov dx, offset str2

int 21h

call nl

mov dx, offset str3

int 21h

ret

main endp

nl proc

push ax

mov ah, 02h

mov dl, 0ah

int 21h

mov dl, 0dh

int 21h

pop ax

ret

nl endp

.data

str1 db "first string", '$'

str2 db "2econd string", '$'

str3 db "third string", '$'

Листинг кода 2 – Текст программы задания 1 формата exe

.model small

.stack 100h

.code

main proc

mov ax, @data

mov ds, ax

mov ah, 09h

mov dx, offset str1

int 21h

call nl

mov dx, offset str2

int 21h

call nl

mov dx, offset str3

int 21h

mov ah, 4Ch

int 21h

main endp

nl proc

push ax

mov ah, 02h

mov dl, 0ah

int 21h

mov dl, 0dh

int 21h

pop ax

ret

nl endp

.data

str1 db "first string", '$'

str2 db "2econd string", '$'

str3 db "third string", '$'

Для сортировки будет использоваться простейший метод пузырька. Также в программе для удобства используется макрос для записи в регистр разницы между содержимым двух меток. Используются процедуры для вывода строки в консоль, определения границ двух слов, сравнение двух слов по границам и обмен двух слов по границам.

Ввод осуществляется через консоль. Программа обрабатывает случай пустого ввода в консоль, а в случае корректного ввода выводит отсортированную строку.

Листинг кода 3 – Текст программы задания 2

.model tiny

.code

diff\_to macro label1, label2, result\_reg

push ax

push bx

mov ax, [label1]

mov bx, [label2]

sub ax, bx

mov result\_reg, ax

pop bx

pop ax

endm

main proc

start:

lea dx, hello\_m

mov ah, 09h

int 21h

call new\_line

mov str, 200

mov ah, 0ah

mov dx, offset str

int 21h

mov al, ' '

mov di, offset str + 2

mov bl, 0

mov cx, 200

sp\_check:

repe scasb

dec di

inc cx

cmp [di], 0dh

je cnt\_end

inc bl

word\_skip:

cmp [di], ' '

je sp\_check

cmp [di], 0dh

je cnt\_end

inc di

dec cx

jmp word\_skip

cnt\_end:

cmp bl, 0 ; bl contains words number

jne preloop

lea dx, e\_str\_m

call printl

call new\_line

jmp start

preloop:

cmp bl, 1

je results

dec bl

mov [w\_cntr], bl

outer\_loop:

mov ch, 0

mov cl, [w\_cntr]

mov dx, 0 ; sets flag

mov di, offset str + 2

inner\_loop:

call cmp\_ws

cmp ax, 1

jne inl1 ; calls find\_ws

mov dx, 1

call w\_swap

inl1:

mov di, [beg\_fst]

call find\_ws

mov di, [end\_fst]

loop inner\_loop

cmp dx, 1

je outer\_loop

results:

lea dx, res\_str

call printl

lea di, str

add di, 2

mov si, offset str

add si, 1

mov al, [si]

mov ah, 0

add di, ax

mov al, '$'

stosb

lea dx, str

add dx, 2

call printl

ret

main endp

find\_ws proc ; set beg\_fst ... end\_snd for words positions

; reg di should be pointed to the current str pos

; doesn't contain any checks

push ax

push cx

mov cx,204

mov al, ' '

repe scasb

dec di

mov offset beg\_fst, di

repne scasb

dec di

mov offset end\_fst, di

repe scasb

dec di

mov offset beg\_snd, di

f\_word\_skip:

cmp [di], ' '

je f\_cont

cmp [di], 0dh

je f\_cont

inc di

jmp f\_word\_skip

f\_cont:

mov offset end\_snd, di

pop cx

pop ax

ret

find\_ws endp

cmp\_ws proc ; return result in ax:

; 1 (first is bigger), -1(second is greater), 0 (strings are the same size)

push bx

push cx

push dx

push si

push di

call find\_ws

mov si, [beg\_fst]

mov di, [beg\_snd]

diff\_to end\_fst, beg\_fst, cx

diff\_to end\_snd, beg\_snd, dx

cmp cx, dx

jae str\_comp

mov cx, dx

str\_comp:

repe cmpsb

ja first\_is\_greater

jb second\_is\_greater

mov ax, 0 ; are equal

jmp done

first\_is\_greater:

mov ax, 1

jmp done

second\_is\_greater:

mov ax, -1

done:

pop di

pop si

pop dx

pop cx

pop bx

ret

cmp\_ws endp

new\_line proc

push ax

push dx

mov ah, 02h

mov dl, 0ah

int 21h

mov dl, 0dh

int 21h

pop dx

pop ax

ret

new\_line endp

printl proc ; requires string pointer in dx

push ax

call new\_line

mov ah, 09h

int 21h

pop ax

ret

printl endp

w\_swap proc ; swap words using find\_ws

push di

push si

push cx

lea di, buf

mov si, [beg\_snd]

diff\_to end\_snd, beg\_snd, cx

rep movsb

diff\_to beg\_snd, end\_fst, cx

mov al, ' '

rep stosb

mov si, [beg\_fst]

diff\_to end\_fst, beg\_fst, cx

rep movsb

diff\_to end\_snd, beg\_fst, cx

mov di, [beg\_fst]

lea si, buf

rep movsb

pop cx

pop si

pop di

ret

w\_swap endp

.data

size equ 200

str db 202 dup (?)

flag db ? ; 0 or 1

beg\_fst dw ?

end\_fst dw ?

beg\_snd dw ?

end\_snd dw ?

w\_cntr db 0

hello\_m db "Enter a string to sort$"

e\_str\_m db "The string can't be empty!$"

res\_str db "Sorted string:$"

buf db ?

# Вывод

В ходе выполнения лабораторной работы были получены навыки написания простейший com и exe программ, обработки строковых данных с использованием инструкций повторений и строковых инструкций. Ознакомились с работой макросов и процедур.