Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет

Лабораторная работа №7

Выполнил студент гр. 150505: Лисичкин Д.А.

Проверил: Туровец Н.О.

Минск 2022

**Тема работы:** Загрузка и выполнение программ. Работа с памятью.

**Цель работы:** Ознакомиться с загрузкой и выполнением программ, рассмотреть работу с памятью.

**Вариант лабораторной работы:** 1. Написать программу, запускающую другую программу N раз (N – число в диапазоне [1, 255]). Имя запускаемой программы задается константой.

**Код программы**

.code

org 100h

start:

call command\_line

str\_to\_num number,numberSize

continue:

mov getNumber, ax

cmp ax, max\_size

ja bad\_input

cmp ax, 1

jl bad\_input

newProgram:

mov ax, cntPrograms

cmp ax, getNumber

jae exit

mov sp, program\_length+100h+200h

mov ah, 4Ah

stack\_shift = program\_length+100h+200h

mov bx, stack\_shift shr 4+1

int 21h

moj kwlv ax, cs

mov word ptr EPB+4, ax

mov word ptr EPB+8, ax

mov word ptr EPB+0Ch, ax

mov ax, 4B00h

mov dx, offset program\_path

mov bx, offset EPB

int 21h

inc cntPrograms

jmp newProgram

exit:

int 20h

command\_line proc

push cx

push si

push di

push ax

xor cx, cx

mov cl, es:[80h]

cmp cl, 0

je no\_command\_line

mov di, 82h

mov si, offset number

read:

mov al, es:[di]

cmp al, 20h

je bad\_input

cmp al, 0Dh

je param\_is\_ended

mov [si], al

inc di

inc numberSize

cmp numberSize,3

jg no\_command\_line

inc si

jmp read

param\_is\_ended:

mov [si], 24h

pop ax

pop di

pop si

pop cx

ret

command\_line endp

no\_command\_line:

cout [exc\_cmd\_message]

jmp exit

cout macro value

push ax

push dx

mov ah, 09h

lea dx, value

int 21h

pop dx

pop ax

endm

str\_to\_num macro number, size

push cx

push dx

push bx

push si

push di

xor ax,ax

xor dx,dx

mov dl,[number]

mov ax,size

lea si,number

mov di,10

mov cx,ax

jcxz bad\_input

xor ax,ax

xor bx,bx

xor dx,dx

mov bl,byte ptr[si]

push bx

cmp bl,'-'

jne loop\_input

jmp bad\_input

endm

loop\_input:

mov bl,[si]

inc si

cmp bl,'0'

jl bad\_input

cmp bl,'9'

jg bad\_input

sub bl,'0'

mul di

jc bad\_input

add ax,bx

jc overflaw

loop loop\_input

jmp end\_input

bad\_input:

xor ax,ax

cout [exc\_cmd\_message]

jmp exit

overflaw:

cout [exc\_overflaw\_message]

jmp exit

end\_input:

pop bx

pop di

pop si

pop bx

pop dx

pop cx

jmp continue

number db 9 dup(0)

numberSize dw 0

getNumber dw ?

max\_size equ 255

cntPrograms dw 0

program\_path db "hi-world.com", 0

EPB dw 0000

dw offset commandline,0

dw 005Ch,0,006Ch, 0

commandline db 125

db " /?"

command\_text db 122 dup (?)

program\_length equ $-start

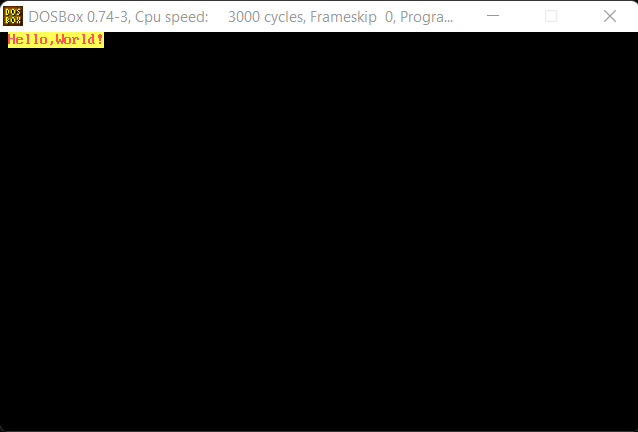
exc\_cmd\_message db "Error !!! N must E [1;255].", '$'

exc\_program\_path\_message db "Program path is not found !!!", '$'

exc\_overflaw\_message db "OWERFLAW !!!", '$'

end start

**Вывод в консоли после компиляции**

****