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 №6

на тему:

**«ЗАГРУЗКА И ВЫПОЛНЕНИЕ ПРОГРАММ. РАБОТА С ПАМЯТЬЮ»**

БГУИР 6-05-0612-02 5

|  |
| --- |
| Выполнила студентка группы 353504  АНТОНОВА Лидия Сергеевна |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 5.** Написать программу, запускающую другую программу с заданными параметрами. Имя запускаемой программы задается константой, а параметры программы задаются в строках текстового файла.

# 2 Выполнение работы

Для написания данной лабораторной работы используется эмулятор emu8086, а также DOSBox для запуска программы. Ниже представлен листинг кода программы.

.model small

.stack 100h

.data

fileName db 32 dup(0)

fileID dw 0

programName db 'test.exe', 0

char db 0

sizeLine db 0

EPB dw 0000

dw offset commandline, 0

dw 005Ch, 0

dw 006Ch, 0

commandline db 0

db 125 dup(0)

msgStart db 'Starting the program...', 0Dh, 0Ah, '$'

msgReadFile db 'Reading parameters from file...', 0Dh, 0Ah, '$'

msgErrorOpenFile db 'Error: Could not open file.', 0Dh, 0Ah, '$'

msgErrorReadFile db 'Error: Could not read file.', 0Dh, 0Ah, '$'

msgLaunching db 'Launching the program...', 0Dh, 0Ah, '$'

msgErrorOpenProgram db "Don't open!", 0Ah, 0Dh, '$'

msgError02h db "ERROR(02h): the file was not found.", 0Ah, 0Dh, '$'

msgError05h db "ERROR(05h): access to the file is denied.", 0Ah, 0Dh, '$'

msgError08h db "ERROR(08h): not enough memory.", 0Ah, 0Dh, '$'

msgError0Ah db "ERROR(0Ah): the wrong environment.", 0Ah, 0Dh, '$'

msgError0Bh db "ERROR(0Bh): incorrect format.", 0Ah, 0Dh, '$'

msgEnter db 0Dh, 0Ah, '$'

dsize = $-commandline

.code

start:

mov ax, @data

mov ds, ax

lea dx, msgStart

call output

call getCmdLine

call openFile

call readFile

call closeFile

call launchProgram

getCmdLine proc

mov cx, 0

mov cl, ES:[80h]

cld

mov di, 81h

mov al, ' '

rep scasb

dec di

lea si, fileName

copy:

mov al, ES:[DI]

cmp al, 0Dh

je endCmd

cmp al, 20h

je endCmd

cmp al, 9h

je endCmd

mov DS:[SI], al

inc di

inc si

jmp copy

endCmd:

mov byte ptr [si], 0

ret

getCmdLine endp

openFile:

mov ah, 3Dh

mov al, 0

lea dx, fileName

int 21h

jc openFileError

mov fileID, ax

ret

openFileError:

lea dx, msgErrorOpenFile

call output

call exit

readFile proc

mov sizeLine, 0

lea si, commandline + 1

mov bx, fileID

mov cx, 1

lea dx, char

nextChar:

mov ah, 3Fh

int 21h

jc exitReadFile

cmp ax, 0

je endOfFile

cmp sizeLine, 125

je endOfFile

inc sizeLine

mov al, char

mov [si], al

inc si

jmp nextChar

endOfFile:

mov byte ptr[si], 0Dh

exitReadFile:

cmp sizeLine, 0

je readFileError

mov al, sizeLine

mov [commandline], al

ret

readFileError:

lea dx, msgErrorReadFile

call output

call exit

readFile endp

closeFile:

mov ah, 3Eh

int 21h

ret

launchProgram proc

mov sp, csize + 100h + 200h

mov ah, 4Ah

mov bx, (csize / 16) + 256 / 16 + (dsize / 16) + 20

int 21h

mov ax, cs

mov word ptr EPB + 4, ds

mov word ptr EPB + 8, cs

mov word ptr EPB + 0Ah, cs

lea dx, msgLaunching

call output

mov ax, @data

mov es, ax

mov ah, 4Bh

mov al, 00h

lea dx, programName

lea bx, EPB

int 21h

jc errorOpen

call exit

launchProgram endp

exit:

mov ax, 4C00h

int 21h

output:

mov ah, 09h

int 21h

ret

errorOpen:

cmp ax, 02h

je error02h

cmp ax, 05h

je error05h

cmp ax, 08h

je error08h

cmp ax, 0Ah

je error0Ah

cmp ax, 0Bh

je error0Bh

lea dx, msgErrorOpenProgram

call output

call exit

error02h:

lea dx, msgError02h

call output

call exit

error05h:

lea dx, msgError05h

call output

call exit

error08h:

lea dx, msgError08h

call output

call exit

error0Ah:

lea dx, msgError0Ah

call output

call exit

error0Bh:

lea dx, msgError0Bh

call output

call exit

csize = $-start

end start

Программа начинается с инициализации переменных и вывода сообщения о старте. Затем она получает имя файла из командной строки и пытается открыть его.

При успешном открытии файла программа считывает параметры и затем закрывает файл. После этого она подготавливает и запускает другой исполняемый файл, передавая ему параметры. В случае ошибок при открытии или чтении файла, программа выводит соответствующие сообщения об ошибках и завершает выполнение. Работа продемонстрировала навыки взаимодействия с файловой системой и обработки ошибок в ассемблере.

На рисунке 1 представлен результат выполнения программы с файлом helloWorld.com.

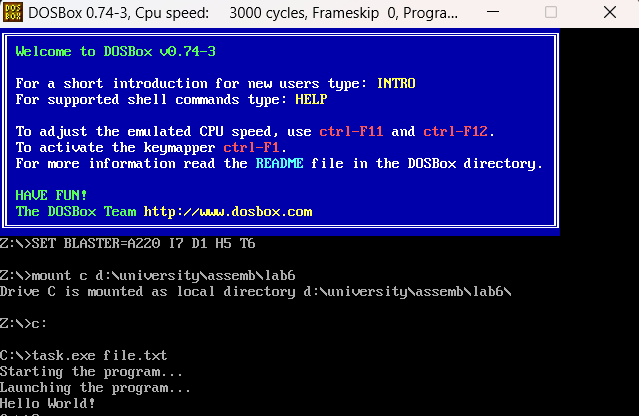


Рисунок 1 – Результат выполнения программы с файлом helloWorld.com

Ниже представлен листинг кода файла helloWorld.asm.

name "helloWorld", 0

.model tiny

.code

org 100h

mov ah, 9

mov dx, offset message

int 21h

ret

message db "Hello World!", '$'

На рисунке 2 представлен результат выполнения программы с файлом string.exe.

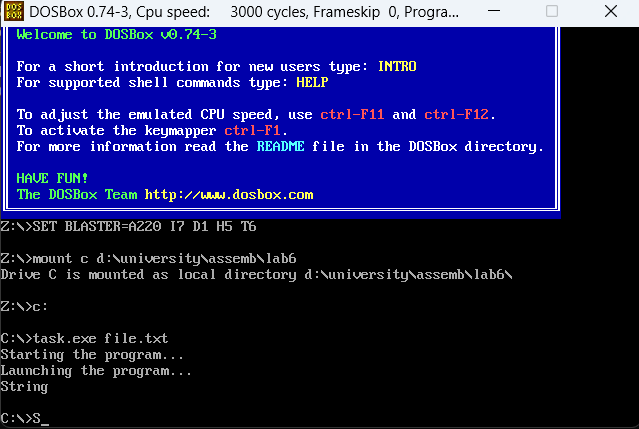


Рисунок 2 – Результат выполнения программы с файлом string.exe

Ниже представлен листинг кода файла string.asm.

.model small

.stack 100h

.data

msg db "String", 0Dh, 0Ah, '$'

.code

mov ax, @data

mov ds, ax

mov ah, 9

mov dx, offset msg

int 21h

mov ax, 4C00h

int 21h

# Вывод

В ходе выполнения лабораторной работы было достигнуто ознакомление с процессом загрузки и выполнения программ в среде DOS. Рассмотрены методы работы с памятью, включая выделение и управление памятью для различных задач.