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 №7

на тему:

**«Загрузка и выполнение программ. Работа с памятью.»**

БГУИР 1-40 04 01

|  |
| --- |
| Выполнил студент группы 253505  СЕНЬКО Никита Святославович |
|  |
| (дата, подпись студента) |
| Проверил ассистент кафедры информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

МИНСК 2023

**Цель работы:** Изучить основные операции обработки файлов на языке Assembler для микроконтроллера Intel 8086.

В рамках выполнения лабораторной работы должно быть выполнено следующее задание:

Написать программу, запускающую другие программы. Список запускаемых программ задается в строках текстового файла.

**Ход работы:** Для выполнения задания был написан программный код на языке Assembler, представленный в листинге 1.

Листинг 1 – Исходный кодосновной программы

.model small

.data

EPB dw 0

cmd\_off dw offset commLine

cmd\_seg dw ?

fcb1 dd 0

fcb2 dd 0

Len dw $-EPB

commLine db 0, ' ', 126 dup(0), '$'

path db "text.txt", 0, "$"

readed\_total dw 0

cmd\_length dw 0

handle dw 0

file dw 0

buffer db 50 dup('$'), '$'

buffer\_prog db 50 dup('$'), '$'

enters db 0Dh, 0Ah,0Dh, 0Ah,'$', '$'

counter dw 0

*;errors*

error\_cmd db 0Ah, 0Dh, "Limit symbols in cmd string!", "$"

error\_not\_found\_mes db 0Ah, 0Dh, "File not found!", "$"

error\_not\_found\_path db 0Ah, 0Dh, "Path not found!", "$"

error\_many\_files\_mes db 0Ah, 0Dh, "Too many files open!", "$"

error\_access\_denied\_mes db 0Ah, 0Dh, "Access is denied!", "$"

error\_incorrect\_access\_mode\_mes db 0Ah, 0Dh, "Incorrect access mode!", "$"

error\_invalid\_id\_mes db 0Ah, 0Dh, "Invalid ID of file!", "$"

rsErr db 0Ah, 0Dh, "Error in resize", "$"

stErr db 0Ah, 0Dh, "Error in start program", "$"

DataSize = $ - EPB

.stack 100h

.code

main:

mov ax, @data

mov ds, ax

mov cmd\_seg, ax

mov cmd\_length, 8

mov cx, cmd\_length

jmp main\_proccess

main\_proccess: *;MAIN*

call openFile

jc error\_badOpen

mov[file], ax

call readFromFile

jc error\_badRead

mov bx, ax

mov byte ptr[commLine], bl

mov readed\_total, ax

call closeFile

jc error\_close

mov si, 0

mov di, 0

mov cx, readed\_total

jmp pre\_loop

error\_close:

jmp error\_invalid\_id

error\_badRead:

jmp badRead

error\_badOpen:

jmp badOpen

pre\_loop:

mov ax, 4A00h

mov bx, ((CodeSize / 16) + 17) + ((DataSize / 16) + 17) + 1

int 21h

jc errResize

loop\_get\_file:

mov al, buffer[si]

cmp al, 0Dh

je start\_prog

mov buffer\_prog[di], al

inc si

inc di

dec cx

jmp loop\_get\_file

start\_prog:

mov buffer\_prog[di], 0

inc di

mov buffer\_prog[di], '$'

mov dx, offset buffer\_prog

mov ah, 09h

int 21h

mov dx, offset enters

mov ah, 02h

int 21h

mov ax, @data

mov es, ax

mov ah, 4Bh

mov al, 00h

mov dx, offset buffer\_prog

mov bx, offset EPB

int 21h

jc errStart

sub cx, counter

cmp cx, 2

je end\_program

add cx, counter

add counter, 2

inc si

inc si

mov di, 0

jmp loop\_get\_file

errStart:

mov ah, 09h

lea dx, stErr

int 21h

jmp end\_program

errResize:

mov ah, 09h

lea dx, rsErr

int 21h

jmp end\_program

end\_program:

mov ah, 4ch

int 21h

openFile proc *;try open file*

push dx

mov dx, offset path

mov ah, 3Dh *;open file*

mov al, 00h *;*

mov cx, 00h

int 21h

mov handle, ax

pop dx

ret

openFile endp

readFromFile proc

mov bx, [file]

mov ah, 3Fh

mov cx, 50

lea dx, buffer

int 21h

ret

readFromFile endp

closeFile proc

mov bx, handle

mov ah, 3Eh

int 21h

ret

closeFile endp

badOpen:

cmp ax, 02h *;not found*

je error\_not\_found

cmp ax, 03h *;path not found*

je error\_path\_not\_found

cmp ax, 04h *;too many open files*

je error\_many\_files

cmp ax, 05h *;access is denied*

je error\_access\_denied

cmp ax, 0Ch *;incorrect access mode*

je error\_incorrect\_access\_mode

badRead:

cmp ax, 05h

je error\_access\_denied

cmp ax, 06h

je error\_invalid\_id

error\_invalid\_id:

mov ax, @data

mov ds, ax

mov dx, offset error\_invalid\_id\_mes

mov ah, 09h

int 21h

jmp end\_program

error\_not\_found:

mov ax, @data

mov ds, ax

mov dx, offset error\_not\_found\_mes

mov ah, 09h

int 21h

jmp end\_program

error\_path\_not\_found:

mov ax, @data

mov ds, ax

mov dx, offset error\_not\_found\_path

mov ah, 09h

int 21h

jmp end\_program

error\_many\_files:

mov ax, @data

mov ds, ax

mov dx, offset error\_many\_files\_mes

mov ah, 09h

int 21h

jmp end\_program

error\_access\_denied:

mov ax, @data

mov ds, ax

mov dx, offset error\_access\_denied\_mes

mov ah, 09h

int 21h

jmp end\_program

error\_incorrect\_access\_mode:

mov ax, @data

mov ds, ax

mov dx, offset error\_incorrect\_access\_mode\_mes

mov ah, 09h

int 21h

jmp end\_program

CodeSize = $ - main

end main

**Вывод:** в ходе выполнения лабораторной работы были изучены основные операции на языке Assembler для микроконтроллера Intel 8086, а также было выполнено предоставленное задание.