

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 253504  
ЖГУТОВ Евгений Дмитриевич

---

(дата, подпись студента)

Проверил ассистент кафедры информатики  
РОМАНЮК Максим Валерьевич

---

(дата, подпись преподавателя)

Минск 2023

**Цель работы:** Задание 1.Вариант 6.Написать программу, запускающую другие программы. Список запускаемых программ задается в строках текстового файла.

**Ход работы:** на рисунке 1 представлены изначальные вводимые данные, на рисунке 2 – результат работы программы.

#### Листинг 1 – Исходный код программы задания 1

```
.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 "test.txt", 0, "$"
readed_total dw 0
cmd_length dw 0
FILE_ID1 dw 0
file dw 0
buffer db 50 dup('$'), '$'
buffer_prog db 50 dup('$'), '$'
enters db 0Dh, 0Ah, 0Dh, 0Ah, '$', '$'
counter dw 0
;errors
error_cmd db 0Ah, 0Dh, "Limit symbols in cmd
string!", "$"
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

```

```

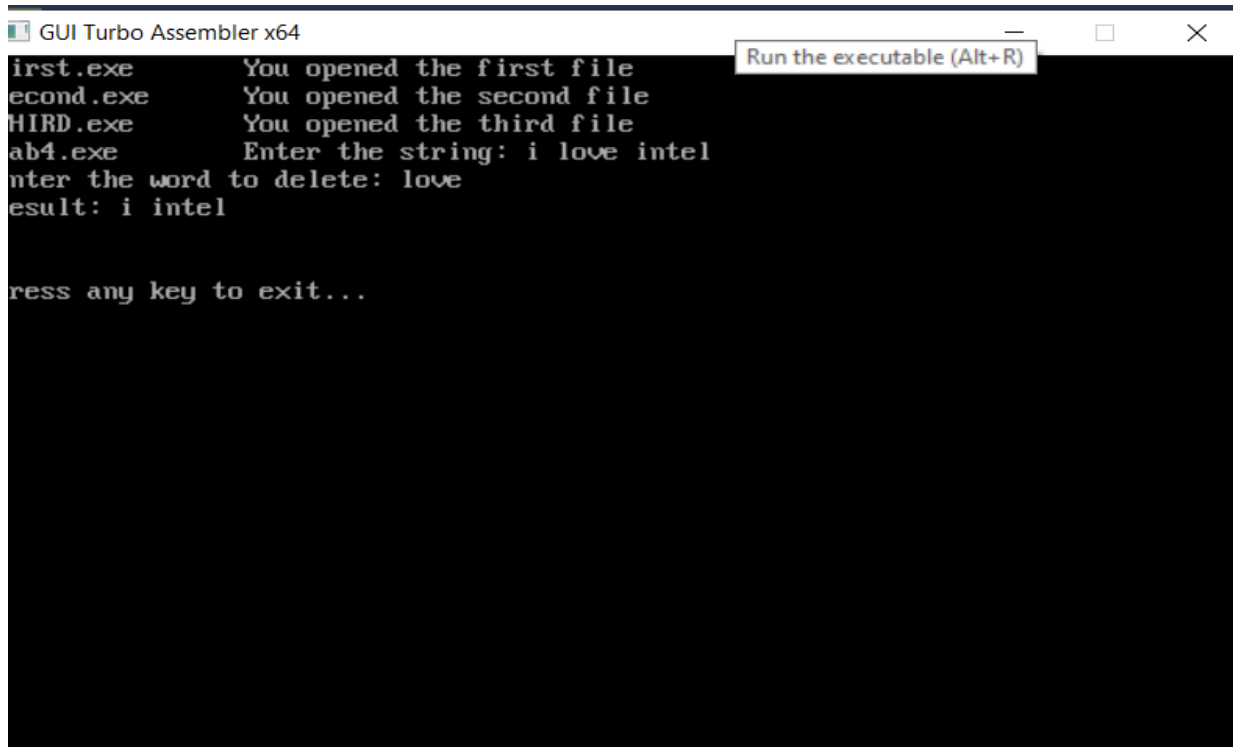
        int 21h
        jmp end_program
end_program:
        mov ah, 4ch
        int 21h
openFile proc                                ;try open file
        push dx
        mov dx, offset path
        mov ah, 3Dh ;open file
        mov al, 00h ;
        mov cx, 00h
        int 21h
        mov FILE_ID1, ax
        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, FILE_ID1
        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

```

A screenshot of a Windows application window titled "GUI Turbo Assembler x64". The window has a standard Windows title bar with minimize, maximize, and close buttons. A tooltip "Run the executable (Alt+R)" is visible over the maximize button. The main area of the window is black with white text. The text shows the execution of a program with several files: "first.exe", "second.exe", and "HIRD.exe" are listed with the message "You opened the file". "ab4.exe" prompts for a string, and the user has entered "i love intel". The program then prompts for a word to delete, and the user has entered "love". The final output is "result: i intel". At the bottom, it says "press any key to exit...".

```
GUI Turbo Assembler x64
first.exe      You opened the first file
second.exe    You opened the second file
HIRD.exe      You opened the third file
ab4.exe       Enter the string: i love intel
Enter the word to delete: love
result: i intel

press any key to exit...
```

Рисунок 1 – Результат работы программы

**Выводы:** В результате лабораторной работы была выполнена одна задача с использованием загрузки и выполнением программ, а также работа с памятью.