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 73

|  |
| --- |
| Выполнила студентка группы 353504  ЛЕБЕДЕВА Милана Валерьевна |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 3.** Написать программу, запускающую другую программу N раз (N – число в диапазоне [1, 255]). Имя запускаемой программы и ее параметры передаются в командной строке.

# 2 Выполнение работы

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

.model small

.data

;VARIABLES

filename db 80 dup(0)

cmdLengthFact dw ?

arguments db 120 dup(0)

EPB dw 0 ;occupied space for EPB

dw offset commandline, 0 ;ptr for CMD

dw 005Ch, 0, 006Ch, 0 ;ptrs for FCB

commandline db 125

db " /?"

size dw ?

;MESSAGES

mess\_noData db "No data.", 0Ah, 0Dh, '$'

mess\_start db "...start...", 0Ah, 0Dh, '$'

messOK db "OK", 0Ah, 0Dh, '$'

mess\_error db "ERROR", 0Ah, 0Dh, '$'

mess\_errorNumber db "The number must be in the range [1; 255].", 0Ah, 0Dh, '$'

mess\_errorOpen db "Don't open!", 0Ah, 0Dh, '$'

mess\_error02h db "ERROR(02h): the file was not found.", 0Ah, 0Dh, '$'

mess\_error05h db "ERROR(05h): access to the file is denied.", 0Ah, 0Dh, '$'

mess\_error08h db "ERROR(08h): not enough memory.", 0Ah, 0Dh, '$'

mess\_error0Ah db "ERROR(0Ah): the wrong environment.", 0Ah, 0Dh, '$'

mess\_error0Bh db "ERROR(0Bh): incorrect format.", 0Ah, 0Dh, '$'

mess\_click db 0Ah,0Dh,"Click...$"

dsize dw $-filename

.stack 256

.code

;MACROS

display macro str

push dx

push ax

lea dx, str

mov ah, 09h

int 21h

pop ax

pop dx

endm

;PROCEDURES

sizeInput proc

lea di, arguments

xor ax, ax

xor bx, bx

parseLoop:

mov bl, [di]

cmp bl, 0Dh

je endParsee

cmp bl, 0

je endParsee

cmp bl, '0'

jb invalidInput

cmp bl, '9'

ja invalidInput

sub bl, '0'

mov cx, 10

mul cx

add ax, bx

inc di

jmp parseLoop

invalidInput:

jmp endParsee

endParsee:

mov size, ax

ret

sizeInput endp

;define fileName and arguments

get\_name proc

push ax

push cx

push di

push si

xor cx, cx

mov cl, es:[80h] ;get count of symbols

mov cmdLengthFact, cx

cmp cl, 0

je endParse

mov di, 82h ;start PSP

lea si, filename

cicle1:

mov al, es:[di]

cmp al, ' '

je end\_get\_name

cmp al, 0Dh

je endParse

mov [si], al

inc di

inc si

jmp cicle1

end\_get\_name:

mov [si], 0

inc di

lea si, arguments

cycle2:

mov al, es:[di]

cmp al, ' '

je skip

cmp al, 0Dh

je endParse

mov [si], al

inc di

inc si

jmp cycle2

skip:

mov al, es:[di]

cmp al, ' '

je skip

cmp al, 0Dh

je endParse

lea si, arguments

mov [si], 0

endParse:

dec si

lea si, arguments

pop si

pop di

pop cx

pop ax

ret

get\_name endp

start:

mov ax, @data

mov ds, ax

;get arguments of CMD

call get\_name

;check CMD

mov ax, cmdLengthFact

cmp ax, 1

jle noData

display mess\_start

;convert to decimal

call sizeInput

;check number

mov ax, size

cmp ax, 1

jl errorNumber

cmp ax, 255

jg errorNumber

;mov ax, 03

;int 10h

;stack setup and memory allocation

mov sp, csize + 100h + 200h

mov ah, 4ah

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

int 21h

jc er ;check flags

mov ax, cs

;fill EPB

mov word ptr EPB+02h, ax

mov word ptr EPB+06h, ax

mov word ptr EPB+0Ah, ax

mov cx, size

openProgram:

;mov ah, 02h

;mov dl, cl

;int 21h

mov ax, 4B00h

lea dx, filename

lea bx, EPB

int 21h

jc erOpen

loop openProgram

click:

display mess\_click

mov ah, 1

int 21h

mov ax, 4C00h

int 21h

noData:

display mess\_noData

jmp click

er:

display mess\_error

jmp click

erOpen:

display mess\_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

error02h:

display mess\_error02h

jmp click

error05h:

display mess\_error05h

jmp click

error08h:

display mess\_error08h

jmp click

error0Ah:

display mess\_error0Ah

jmp click

error0Bh:

display mess\_error0Bh

jmp click

errorNumber:

display mess\_errorNumber

jmp click

csize dw $-start

end start

Для корректной обработки результатов используются соответствующие сообщения и переменные. mess\_noData, mess\_errorNumber, mess\_start и т.д. – это строки, которые используются для вывода различных сообщений в программе. cmdLengthFact содержит фактическую длину командной строки, filename хранит имя файла, указанное в командной строке, arguments хранит аргументы командной строки, EPB зарезервированное место для EPB (Environmental Parameter Block), commandline длина командной строки, size переменная для хранения числа запусков.

. На рисунке 1 представлен результат выполнения программы с файлом hi-world.com.

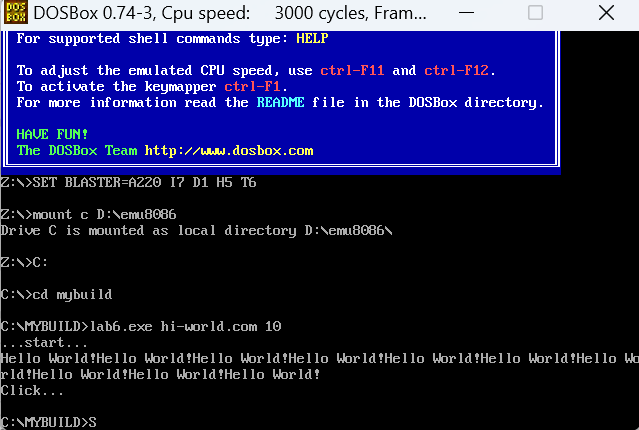


Рисунок 1 – Результат выполнения программы с файлом hi-world.com

Ниже представлен листинг кода файла hi-world.asm.

name "hi-world"

.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 1.exe.

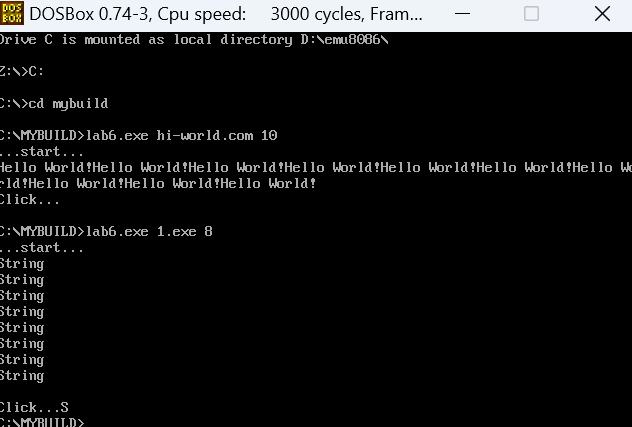


Рисунок 2 – Результат выполнения программы с файлом 1.exe

Ниже представлен листинг кода файла 1.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

На рисунке 3 представлен результат выполнения обработки ошибок.

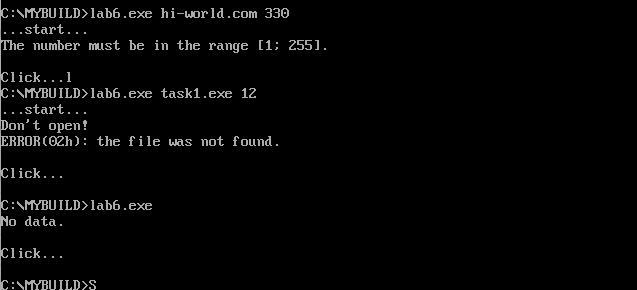


Рисунок 3 – Результат выполнения обработки ошибок

# 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.