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 67

|  |
| --- |
| Выполнил студент группы 353503  КОХАН Артём Игоревич |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 7.** Написать программу, последовательно запускающую программы, которые расположены в заданном каталоге.

# 2 Выполнение работы

Для того, чтобы запускать файлы формата .exe нам понадобиться область dta, которая используется для временного хранения информации о файле, для выполнения операций чтения и записи. При поиске .exe файлов, информация о найденных файлах записывается в область DTA, что позволяет последовательно запускать .exe файлы. Занимает эта область 44 байта. 0-1 байты: Зарезервированы. 2-3 байты: Атрибуты файла. 4-7 байты: Время и дата последнего изменения файла. 8-15 байты: Размер файла. 16-31 байты: Имя и расширение файла (8 + 3 байта). 32-43 байты: Используются для текущего блока и текущего числа записей.

Также ключевую роль играет EPB (Execution Parameter Block), которая используется для хранения параметров, передаваемых в исполняемые программы. Это структура данных, обеспечивающая выполнение программ DOS, и включающая информацию о командной строке и окружении, где dw 005Ch, 0, 006Ch, 0: Зарезервировано для параметров FCB (File Control Block). FCB, в основном, используется для выполнения операций с файлами на низком уровне.

Листинг кода 1 – Текст программы задания 1

.model small

.stack 100h

.data

cmd\_size db ?

maxCMDSize equ 127

cmd\_text db maxCMDSize + 2 dup(0)

folderPath db maxCMDSize + 2 dup(0)

DTAsize equ 2Ch

DTAblock db DTAsize dup(0)

startText db "Program is started", '$'

endText db "Program is ended", '$'

reallocErrorMessage db "Bad relocation memory. Error code: ", '$'

runEXEErrorText db "Error running other program. Error code: ", '$'

badFolderErrorText db "Open directory error", '$'

EPBstruct dw 0

dw offset line,0

dw 005Ch, 0, 006Ch, 0

line db 125

db " /?"

line\_text db 122 dup(?)

EPBlen dw $ - EPBstruct

extensionEXEFile db "\*.exe", 0

DataLength = $ - cmd\_size

.code

newline MACRO

push ax

push dx

mov dl, 10

mov ah, 02h

int 21h

mov dl, 13

mov ah, 02h

int 21h

pop dx

pop ax

ENDM

println MACRO info

push ax

push dx

mov ah, 09h

mov dx, offset info

int 21h

newline

pop dx

pop ax

ENDM

printErrorCode MACRO

add al, '0'

mov dl, al

mov ah, 06h

int 21h

newline

ENDM

main:

mov ah, 4Ah

mov bx, ((CodeLength / 16) + 1) + ((DataLength / 16) + 1) + 32

int 21h

jnc startOfMainProgram

println reallocErrorMessage

printErrorCode

mov ax, 1

jmp endMain

startOfMainProgram:

mov ax, @data

mov es, ax

xor ch, ch

mov cl, ds:[80h]

mov cmd\_size, cl

mov bh, cl

dec bh

mov si, 82h

mov di, offset cmd\_text

rep movsb

mov ds, ax

println startText

mov cmd\_size, bh

call parseCMD

mov ah, 3Bh

mov dx, offset folderPath

int 21h

jc openDirectoryError

call findFirstFile

cmp ax, 0

jne endMain

call runEXE

cmp ax, 0

jne endMain

runFile:

call findNextFile

cmp ax, 0

jne endMain

call runEXE

cmp ax, 0

jne endMain

jmp runFile

openDirectoryError:

println badFolderErrorText

endMain:

println endText

mov ah, 4Ch

int 21h

parseCMD PROC

push bx

push cx

push dx

mov si, offset cmd\_text

mov di, offset folderPath

mov cl, cmd\_size

xor ch,ch

rep movsb

pop dx

pop cx

pop bx

ret

parseCMD ENDP

runEXE PROC

push bx

push dx

mov ax, 4B00h

mov bx, offset EPBstruct

mov dx, offset DTAblock + 1Eh

int 21h

jnc runEXEAllGood

println runEXEErrorText

printErrorCode

mov ax, 1

jmp runEXEEnd

runEXEAllGood:

mov ax, 0

runEXEEnd:

pop dx

pop bx

ret

runEXE ENDP

installDTA PROC

mov ah,1Ah

mov dx, offset DTAblock

int 21h

ret

installDTA ENDP

findFirstFile PROC

call installDTA

mov ah,4Eh

xor cx,cx

mov dx, offset extensionEXEFile

int 21h

jnc findFirstFileAllGood

mov ax, 1

jmp findFirstFileEnd

findFirstFileAllGood:

mov ax, 0

findFirstFileEnd:

ret

findFirstFile ENDP

findNextFile PROC

call installDTA

mov ah,4Fh

mov dx, offset DTAblock

int 21h

jnc findNextFileAllGood

mov ax, 1

jmp findNextFileEnd

findNextFileAllGood:

mov ax, 0

findNextFileEnd:

ret

findNextFile ENDP

CodeLength = $ - main

end main

# Вывод

В ходе выполнения лабораторной работы с программным обеспечением, предназначенным для сборки, отладки и запуска программ на языке ассемблера ознакомились с загрузкой и выполнением программ, рассмотрена работа с памятью.