Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет

Лабораторная работа №5

Выполнил студент гр. 150505: Лисичкин Д.А.

Проверил: Туровец Н.О.

Минск 2022

**Тема работы:** Работа с файлами.

**Цель работы:** Ознакомиться с основными операциями обработки файлов, получить понятие о работе с параметрами командной строки.

**Вариант лабораторной работы:** 11.В выходной файл поместить только те строки входного файла, которые содержат все указанные символы.

**Код программы**

.model small

.stack 100h

.data

startDX dw 0

tempDX dw 0

flagTemp dw 0

flagStart dw 0

maxCMDSize equ 127

cmd\_size db ?

cmd\_text db maxCMDSize + 2 dup(0)

sourcePath db 129 dup (0)

tempSourcePath db 128 dup (0)

destinationPath db "output.txt",0

extension db "txt"

point2 db '.'

buf db 0

sourceID dw 0

destinationID dw 0

newLineSymbol equ 0Dh

returnSymbol equ 0Ah

endl equ 0

enteredString db 200 dup("$")

enteredStringSize dw 0

startProcessing db "Processing started", '$'

startText db "Program is started", '$'

badCMDArgsMessage db "Bad command-line arguments.", '$'

badSourceText db "Open error", '$'

fileNotFoundText db "File not found", '$'

endText db 0Dh,0Ah,"Program is ended", '$'

errorReadSourceText db "Error reading from source file", '$'

.code

cin MACRO string

push ax

push dx

lea dx, string

mov ah, 0Ah

int 21h

pop dx

pop ax

endm

cout MACRO info

push ax

push dx

mov ah, 09h

lea dx, info

int 21h

mov dl, 0Ah

mov ah, 02h

int 21h

mov dl, 0Dh

mov ah, 02h

int 21h

pop dx

pop ax

ENDM

strcpy MACRO destination, source, count

push cx

push di

push si

xor cx, cx

mov cl, count

lea si, source

lea di, destination

rep movsb

pop si

pop di

pop cx

ENDM

incrementTempPos MACRO num

add tempDX, num

jo overflowTempPos

jmp endIncrementTempPos

overflowTempPos:

inc flagTemp

add tempDX, 32769

jmp endIncrementTempPos

endIncrementTempPos:

endm

incrementStartPos proc

push ax

mov ax, tempDX

add startDX, ax

jo overflow

jmp endIncrement

overflow:

inc flagStart

add startDX, 32769

endIncrement:

mov ax, flagTemp

add flagStart, ax

pop ax

ret

endp

fseekCurrent MACRO settingPos

push ax

push cx

push dx

mov ah, 42h

mov al, 1

mov cx, 0

mov dx, settingPos

int 21h

pop dx

pop cx

pop ax

ENDM

fseek MACRO fseekPos

push ax

push cx

push dx

mov ah, 42h

mov al, 0

mov cx, 0

mov dx, fseekPos

int 21h

pop dx

pop cx

pop ax

ENDM

setPointer proc

push cx

push bx

mov bx, sourceID

fseek startDX

cmp flagStart, 0

je endSetPos

xor cx, cx

mov cx, flagStart

setPos1:

mov bx, sourceID

fseekCurrent 32767

loop setPos1

endSetPos:

pop bx

pop cx

ret

endp

main:

mov ax, @data

mov es, ax

xor ch, ch

mov cl, ds:[80h]

mov bl, cl

mov cmd\_size, cl

dec bl

mov si, 81h

lea di, tempSourcePath

rep movsb

mov ds, ax

mov cmd\_size, bl

mov cl, bl

lea di, cmd\_text

lea si, tempSourcePath

inc si

rep movsb

cout startText

call parseCMD

cmp ax, 0

jne endMain

call openFiles

cmp ax, 0

jne endMain

cin enteredString

xor ax, ax

mov al, [enteredString+1]

mov enteredStringSize, ax

cmp enteredStringSize, 0

je endMain

cout startProcessing

call processingFile

endMain:

cout endText

mov ah, 4Ch

int 21h

parseCMD proc

xor ax, ax

xor cx, cx

cmp cmd\_size, 0

je notFound

mov cl, cmd\_size

lea di, cmd\_text

mov al, cmd\_size

add di, ax

dec di

findPoint:

mov al, '.'

mov bl, [di]

cmp al, bl

je pointFound

dec di

loop findPoint

notFound:

cout badCMDArgsMessage

mov ax, 1

ret

pointFound:

mov al, cmd\_size

sub ax, cx

cmp ax, 3

jne notFound

xor ax, ax

lea di, cmd\_text

lea si, extension

add di, cx

mov cx, 3

repe cmpsb

jne notFound

strcpy sourcePath, cmd\_text, cmd\_size cmd

mov ax, 0

ret

endp

openFiles PROC

push bx

push dx

push si

mov ah, 3Dh

mov al, 02h

lea dx, sourcePath

int 21h

jb badOpenSource

mov sourceID, ax

mov ah, 3Ch

xor cx, cx

lea dx, destinationPath

int 21h

jb badOpenSource

mov ah, 3Dh

mov al, 02h

lea dx, destinationPath

int 21h

jb badOpenSource

mov destinationID, ax

mov ax, 0

jmp endOpenProc

badOpenSource:

cout badSourceText

cmp ax, 02h

jne errorFound

cout fileNotFoundText

jmp errorFound

errorFound:

mov ax, 1

endOpenProc:

pop si

pop dx

pop bx

ret

ENDP

processingFile proc

for1:

mov tempDX, 0

mov flagTemp, 0

mov bx, sourceID

call setPointer

lea si, enteredString

add si, 2

for2:

call readSymbolFromFile

incrementTempPos 1

cmp ax, 0

je endFileGG

cmp [buf], 0

je endFileGG

cmp [buf], returnSymbol

je endString

cmp [buf], newLineSymbol

je endString

cmp [buf], endl

je endString

xor ax, ax

xor bx, bx

mov al, buf

mov bl, [si]

cmp al, bl

je doSomething

jmp for2

endString:

call incrementStartPos

jmp for1

doSomething:

inc si

xor bx, bx

mov bl, [si]

cmp bl, newLineSymbol

je stringUdov

cmp bl, returnSymbol

je stringUdov

cmp bl, endl

je stringUdov

mov tempDX, 0

mov flagTemp, 0

call setPointer

jmp for2

stringUdov:

call writeStr

jmp for1

endFileGG:

ret

endp

writeStr proc

mov bx, sourceID

call setPointer

mov bx, destinationID

mov tempDX, 1

mov flagTemp, 0

while1:

call readSymbolFromFile

call incrementStartPos

cmp ax, 0

je endAll

cmp [buf], returnSymbol

je endWrite

cmp [buf], endl

je endAll

mov ah, 40h

mov cx, 1

lea dx, buf

int 21h

jmp while1

endWrite:

mov ah, 40h

mov cx, 1

lea dx, buf

int 21h

endAll:

ret

endp

readSymbolFromFile proc

push bx

push dx

mov ah, 3Fh

mov bx, sourceID

mov cx, 1

lea dx, buf

int 21h

jnb successfullyRead

cout errorReadSourceText

mov ax, 0

successfullyRead:

pop dx

pop bx

ret

endp

end main

**Вывод в консоли после компиляции**

