Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Конструирование программ

Отчёт по лабораторной работе №8

РАБОТА С ФАЙЛАМИ

Студент Киселёва Е. А.

Принимающий Романюк М.В.

Минск 2022

**СОДЕРЖАНИЕ**

[1 Цель работы 3](file:///C:\Users\Acer\AppData\Local\Temp\Rar$DIa29864.36103\КахновскийЕС_КПрог_Lab4_Asma.docx#_Toc118909172)

[2 Задание 4](file:///C:\Users\Acer\AppData\Local\Temp\Rar$DIa29864.36103\КахновскийЕС_КПрог_Lab4_Asma.docx#_Toc118909173)

[3 Результат выполнения 13](file:///C:\Users\Acer\AppData\Local\Temp\Rar$DIa29864.36103\КахновскийЕС_КПрог_Lab4_Asma.docx#_Toc118909174)

[4 Вывод 15](file:///C:\Users\Acer\AppData\Local\Temp\Rar$DIa29864.36103\КахновскийЕС_КПрог_Lab4_Asma.docx#_Toc118909175)

**1 ЦЕЛЬ РАБОТЫ**

Ознакомиться с основными операциями обработки файлов.

Получить понятие о работе с параметрами командной строки.

2 ЗАДАНИЕ

Вариант 7. Подсчитать число строк в файле, в которых есть указанное слово.

Листинг 1 ­– Код программы

; Распечатать строку [Макрос]

printStr macro str

push ax

push dx

mov ah,09h ; Номер функции DOS

lea dx,str ; Смещение str в DX

int 21h ; 21 прерывание

pop dx

pop ax

endm

; Распечатать символ [Макрос]

printSymb macro ascii

mov ah,06h

mov dl,ascii

int 21h

endm

; Распечатать строку по смещению [Макрос]

printStrByOffset macro offset

push ax

push dx

mov ah,09h ; Номер функции DOS

mov dx,offset ; Смещение в DX

int 21h ; 21 прерывание

pop dx

pop ax

endm

; Открыть файл [Макрос]

openFile MACRO nameOffset, id

mov dx,nameOffset ; Адрес ASCIZ-строки с именем файла

mov ah,3Dh ; Функция DOS 3Dh (открыть существующий файл)

mov al,02h ; 02 - для чтения и записи

int 21h ; Открыть файл

mov id,ax ; Сохранить дескриптор

jc openErrorLabel

endm

; Создать файл [Макрос]

createFile MACRO nameOffset, id

mov ah,3Ch ; Функция DOS 3Ch (создать файл)

mov ch,00100000b ; Атрибут файла

mov dx,nameOffset ; Адрес ASCIZ-строки с именем файла

int 21h ; Создать файл

mov id,ax ; Сохранить дескриптор

jc createErrorLabel

endm

; Считать символы [Макрос]

readChars MACRO id, number, buf

mov ah,3Fh ; Функция DOS (чтение из файла)

mov bx,id ; Загрузка дескриптора

mov cx,number ; Количество считываемых символов

lea dx,buf ; Загрузка буфера

int 21h ; Считать данные

jc readingErrorLabel

endm

; Записать символы [Макрос]

writeChars MACRO id, number, buf

mov ah,40h ; Функция DOS (запись в файл)

mov bx,id ; Дескриптор

mov cx,number ; Количество записываемых символов

lea dx,buf ; Буфер

int 21h ; Записать данные

jc writingErrorLabel

endm

; Установить курсор [Макрос]

setCursor MACRO id, distance, type

mov ah,42h ; Функция DOS (переместить указатель чтения/записи)

mov bx,id ; Идентификатор

mov cx,0h ;

mov dx,distance ;

mov al,type ; Относительно чего

int 21h ; Переместить указатель

jc cursorErrorLabel

endm

; Закрыть файл [Макрос]

closeFile MACRO id

mov bx,id ; Дескриптор

mov ah,3Eh ; Функция DOS 3Eh

int 21h ; Закрыть файл

jc closeErrorLabel

endm

; Удалить файл [Макрос]

deleteFile MACRO nameOffset, labelGood, labelBad

mov ah,41h ; Функция DOS 41h (удалить файл)

mov dx,nameOffset ; Адрес ASCIZ-строки с именем файла

int 21h ; Удалить файл

jc deleteErrorLabel

endm

.model tiny

.code

org 100h

START:

mov end\_SP,sp ; Сохранение SP mov cx,charsRead ; Количество считанных символов

mov al,0h ; Символ NULL

lea di,buffer ; Смещение буфера

repne scasb ; Сравниваем, пока не равны

jnz writeCharsInTempFile ; ZF <> 0?

; Действие, если есть символ NULL

dec charsOfStringInDose ; charsOfStringInDose--

mov isEndOfString,01h ; Конец строки

mov isEndOfFile,01h ; Конец файла

jmp writeCharsInTempFile

; Действие, если есть символ 0Dh

Exists0Dh:

; Определения числа символов строки в порции

; charsOfStringInDose = charsRead - CX + 1

mov dx,charsRead

mov charsOfStringInDose,dx

sub charsOfStringInDose,cx

inc charsOfStringInDose

mov isEndOfString,01h

; Кейс когда .. .. .. .. 0Dh, порция закончилась

; то есть в порцию не влезло 0Ah

mov bx,charsOfStringInDose

mov buffer[bx-1],0Ah

; Запись символов во временный файл

writeCharsInTempFile:

writeChars temp\_file\_ID,charsOfStringInDose,buffer ; Записать символы

mov dx,charsOfStringInDose ; charactersInString +=

add charsInString,dx ; += charOfStringInDose

; Проверка, конец ли строки

cmp isEndOfString,0h

je notEndOfString

; Действие, если в порции есть конец строки

endOfString:

; Установка курсора в source\_file

mov bx,charsRead

mov cursorOffset,bx

mov bx,charsOfStringInDose

sub cursorOffset,bx ; cursorOffset =

neg cursorOffset ; = -(charsRead - charsOfStringInDose)

setCursor source\_file\_ID,cursorOffset,01h ; Установить курсор

; Проверка, подходит ли строка из файла под условие

; "не содержать ни одного символа из заданных"

setCursor temp\_file\_ID,00h,00h ; Курсор на начало файла

mov dx,charsInString ; charsInStringCopy =

mov charsInStringCopy,dx ; charsInString

mov dx,buffer\_dose ; Сравнить

cmp charsInStringCopy,dx ; charsInStringCopy и buffer\_dose

jbe skipCheckCycle ; if charsInStringCopy <= buffer\_dose

; Обработка ситуации, когда последняя строка

; была удалена => от нее осталась пустая строка

; (от символов 0Dh,0Ah предыдущей строки)

cmp lastStringWasDeleted,01d ; Если не была удалена

jne exitSuccessful ; Выход

setCursor destination\_file\_ID,00h,00h ; Установить курсор на конец файла

readChars destination\_file\_ID,02h,buffer ; Считать два символа

cmp ax,02h ; AX - число считанных

jb exitSuccessful ; AX < 2 => выход

mov cursorOffset,02h ;

neg cursorOffset ; cursorOffset = -2

setCursor destination\_file\_ID,cursorOffset,02h ; Установить курсор за 2 символа до конца

readChars destination\_file\_ID,02h,buffer ; Считать в буфер два символа

cmp buffer,0Dh ; Первый считанный 0Dh?

jne exitSuccessful ; Не 0Dh => выход успешный

lea bx,buffer ; Смещение буфера в bx

mov [bx],' ' ; Заполнение первого и второго

mov [bx + 1],' ' ; элемента буфера

setCursor destination\_file\_ID,cursorOffset,02h ; Установить курсор за 2 символа до конца

writeChars destination\_file\_ID,02h,buffer ; Записать 2 символа из буфера в файл

jmp exitSuccessful ; Выход учпешный

; Обработка ошибок

openErrorLabel:

printStr openError

call errorHandler

jmp exit

createErrorLabel:

printStr createError

call errorHandler

jmp exit

readingErrorLabel:

printStr readingError

call errorHandler

jmp exit

writingErrorLabel:

printStr writingError

call errorHandler

jmp exit

cursorErrorLabel:

printStr cursorError

call errorHandler

jmp exit

closeErrorLabel:

printStr closeError

call errorHandler

jmp exit

deleteErrorLabel:

printStr deleteError

call errorHandler

jmp exit

; Завершение программы

exitSuccessful:

printStr successfulCompleted

exit:

mov sp,end\_SP ; Восстановить SP

printStr New\_Line

printStr Result\_Info\_String

Print\_Number Result

ret ; Завершение COM-программы

Inc\_Result:

inc Result

mov contains, 0d

jmp resetTempfile

; [Процедура] – результат в contains

; Есть ли символы в порции строки

checkBuffer PROC near

; mov contains,0h

mov si,0h ; SI = 0 [счетчик]

dec si

for1:

inc si

mov isFor1\_Or\_2, 1

cmp contains, 1d

je Exit\_Loop

cmp si,bufferNumberCheck

je Exit\_Loop

mov dx,0h ; DX = 0

mov dl,buffer[si] ; DX = текущий символ из порции

; push si ; Сохранение SI

; mov si,0h ; SI = 0 [счетчик]

mov Count\_For\_FindString, 0d

mov Count\_MainString\_Copy, si

push dx

mov dx,Size\_FindString

mov Size\_FindString\_Copy, dx

pop dx

for2:

mov Checking\_Symbol, dl

Start\_For2:

push si

mov si,Count\_For\_FindString

; cmp dl,Find\_String[Count\_For\_FindString]; Сравнение символа из порции с

push dx

mov dl,Checking\_Symbol

cmp dl,Find\_String[si];

pop dx

jne not\_equals ; символом из набора символов

;-----Если символы равны----------------------------

mov isFor1\_Or\_2, 2

pop si

mov contains,01h ; contains = 1

inc Count\_MainString\_Copy

inc Count\_For\_FindString

push si

mov si,Count\_MainString\_Copy

; mov dl, Find\_String[Count\_MainString\_Copy]

push dx

; mov dl,Checking\_Symbol

mov dl, buffer[si]

; mov dl, Find\_String[si]

mov Checking\_Symbol,dl

pop dx

pop si

dec Size\_FindString\_Copy ; Size-- искомого слова

cmp Size\_FindString\_Copy,0 ; Сравнение

;jz for1

je for1 ;

jmp Start\_For2

;--------------------------------------------------

; Не равен

not\_equals:

cmp isFor1\_Or\_2, 1

je not\_eq\_For1

mov contains, 0d

pop si

dec Size\_FindString\_Copy ; Size-- искомого слова

cmp Size\_FindString\_Copy,0 ; Сравнение

;jz for1

je for1 ;

;jmp for1

jmp Start\_For2

not\_eq\_For1:

mov contains, 0d

pop si

jmp for1

Exit\_Loop:

ret

checkBuffer endp

; [Макрос] – есть ли символы в порции строки

checkBufferMacro MACRO number

mov dx,number

mov bufferNumberCheck,dx

call checkBuffer

endm

;====================Процедура вывода числа====================

Print\_Number\_Proc PROC near

mov cx,0h ; Количество цифр у числа

; Добавление цифр в тек

pushDigitsWhile:

mov dx,0h

mov ax,Number\_General ; AX = число

div ten ; DX:AX / ten = DX:AX

mov Number\_General,ax ; numberGeneral уже /= 10

add dx,48d ; DX += 48 (перевод в ASCII)

push dx ; DX = текущая цифра числа

inc cx ; CX ++

; Условие выхода из цикла

cmp Number\_General,0h

jne pushDigitsWhile

; Доставание из стека и печать цифр

printDigitsWhile:

mov dx,0h

mov ax,Number\_General ; AX = число

div ten ; DX:AX / ten = DX:AX

mov Number\_General,ax ; numberGeneral уже /= 10

; Цифра числа DX

; прерывание берет ASCII символа из DL

pop dx

mov ah,06h

int 21h

; Условие выхода из цикла

dec cx

cmp cx,0h

jne printDigitsWhile

ret

Print\_Number\_Proc endp

; printStrByOffset paramOffset ; Распечатать параметр Checking\_Symbol db ; проверяемый символ

; Для алгоритма перевода числа

TEN dw 10d

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

end\_SP dw 0 ; Адрес SP для ret в конце

;source\_file\_name db 'd:\data.txt',0 ; Исходный файл

source\_file\_ID dw 0 ; Дескриптор

source\_file\_offset dw 0

source\_file\_error db 0Ah, 0Dh, 'Error opening the source file$'

;destination\_file\_name db 'd:\result.txt',0 ; Файл с результатом

destination\_file\_ID dw 0 ; Дескриптор

destination\_file\_offset dw 0

temp\_file\_name db 'temp.txt',0 ; Временный файл

temp\_file\_ID dw 0 ; Дескриптор

temp\_file\_offset dw 0

openError db 0Ah, 0Dh, 'File opening error$'

createError db 0Ah, 0Dh, 'File creation error$'

readingError db 0Ah, 0Dh, 'Reading error$'

writingError db 0Ah, 0Dh, 'Writing error$'

cursorError db 0Ah, 0Dh, 'Cursor installation error$'

closeError db 0Ah, 0Dh, 'File closing error$'

deleteError db 0Ah, 0Dh, 'File deletion error$'

error02hMsg db 0Ah, 0Dh, '02h - file not found$'

error03hMsg db 0Ah, 0Dh, '03h - path not found$'

error04hMsg db 0Ah, 0Dh, '04h - too much opened files$'

error05hMsg db 0Ah, 0Dh, '05h - access denied$'

error06hMsg db 0Ah, 0Dh, '06h - invalid ID$'

error0ChMsg db 0Ah, 0Dh, '0Ch - invalid access mode$'

error0FhMsg db 0Ah, 0Dh, '0Fh - a nonexistent disk is specified$'

error10hMsg db 0Ah, 0Dh, '10h - the current directory cannot be deleted$'

error12hMsg db 0Ah, 0Dh, '12h - invalid access mode$'

error50hMsg db 0Ah, 0Dh, '50h - the file already exists$'

started db 0Ah, 0Dh, 'The program has been started$'

successfulCompleted db 0Ah, 0Dh, 'END PROGRAMM$'

buffer db 100 DUP(0) ; Буфер данных

buffer\_dose dw 50d ; Размер порции

isEndOfFile db 0 ; [bool] – конец ли файла

isEndOfString db 0 ; [bool] – конец ли строки

charsRead dw 0 ; Считано символов

charsOfStringInDose dw 0 ; Символов строки в порции

charsInString dw 0 ; Символов в строке

cursorOffset dw 0 ; Смещение курсора

;symbols db '0123456789',0 ; Символы, которых быть не дложно

symbolsOffset dw 0 ; Смещение на символы, которых быть не должно

bufferNumberCheck dw 0 ; Количество символов из буфера для проверки

charsInStringCopy dw 0 ; Символов в строке (копия)

contains db 0 ; [bool] – есть ли в первых charsOfStringInDose

; символах из buffer 1 из символов symbols

lastStringWasDeleted db 0 ; [bool] - последняя стркоа была удалена

paramsNumber db 0 ; Число параметров командной стркои

paramOffset dw 0 ; Смещение параметра

toNewLine db 0Ah, 0Dh, '$'

paramsMessage db 'Command line parameters:$'

paramsYourMessage db 'Your parameters:$'

paramsError db 'Please, enter THREE PARAMETERS$' ;

paramsFormat db '1 - source file name (exists)',0Ah,0Dh,'2 - destination file name (will be created or overwritten)',0Ah,0Dh,'3 - character set (according to the task condition)$'

end START

3 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

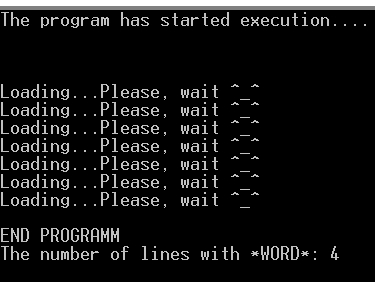
****

Рисунок 1 – Результат работы программы по слову “cat”

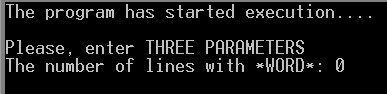


Рисунок 2 – Обработка некорректных параметров командной строки

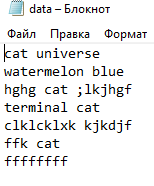


Рисунок 3 – Входной файл

4 ВЫВОД

Таким образом, была создана программа, которая позволяет подсчитать количество строк с заданным словом в файле, данное кол-во выводится пользователю на экран.