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:

**«Работа с файлами.»**

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

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

Листинг 1 – Исходный код программы задания 1

.model small

.stack 100h

.data

crlf db 0Dh, 0Ah, '$'

word db 10, 0, 15 dup('$'), '$'

buf db 2, 0, 2 dup('$'), '$'

msg\_prompt\_to\_input db 'Enter the word to search', 0Dh, 0Ah, '$'

msg\_result db 0Dh, 0Ah, 'Count of strings: $'

msg\_bad db 0Dh, 0Ah, 'Buffer overflow$'

msg\_bad\_args db 'Error: Unable to parse command line arguments', 0Dh, 0Ah, '$'

msg\_empty\_args db 'Error: No command line arguments provided', 0Dh, 0Ah, '$'

msg\_error db 0Dh, 0Ah, 'Error', 0Dh, 0Ah, '$'

word\_capacity equ 50

word\_buffer db word\_capacity + 2 dup(0)

cmd\_capacity equ 127

cmd\_length db ?

cmd\_text db cmd\_capacity dup('$')

file\_path db cmd\_capacity dup('$')

.code

is\_empty macro str, is\_0

push si

lea si, str

call strlen

pop si

cmp ax, 0

je is\_0

endm

output macro str

push ax

push dx

lea dx, str

mov ah, 9

int 21h

pop dx

pop ax

endm

input macro str

push bx

push cx

push dx

again:

mov ah, 0Ah

lea dx, str

int 21h

xor ax, ax

xor cx, cx

mov cl, [word + 1]

cmp cl, 0

je again

pop dx

pop cx

pop bx

endm

puti macro

local put1

local put2

local ex

push ax

push cx

push -1

mov cx, 10

put1:

xor dx, dx

xor ah, ah

div cl

mov dl, ah

push dx

cmp al, 0

jne put1

mov ah, 2

put2:

pop dx

cmp dx, -1

je ex

add dl, '0'

int 21h

jmp put2

ex:

mov dl, ' '

int 21h

pop cx

pop ax

endm

fopen macro

lea dx, file\_path

mov ah, 3Dh

mov al, 00h

int 21h

jc exit

mov bx, ax

endm

fclose macro

mov ah, 3Eh

int 21h

endm

fread macro

local continue

push ax

push cx

push dx

mov cx, 1

lea dx, buf

mov ah, 3Fh

int 21h

jc exit

mov cx, ax

test cx, cx

jnz continue

fclose

jmp good\_exit

continue:

pop dx

pop cx

pop ax

endm

parse\_cmd\_text proc

push bx

push cx

push dx

mov cl, cmd\_length

xor ch, ch

lea si, cmd\_text

lea di, file\_path

call to\_asciiz

is\_empty file\_path, bad\_cmd\_args

lea di, word\_buffer

call to\_asciiz

is\_empty word\_buffer, good\_cmd\_args

bad\_cmd\_args:

output msg\_bad\_args

mov ax, 1

jmp end\_parse\_cmd\_text

good\_cmd\_args:

mov ax, 0

end\_parse\_cmd\_text:

pop bx

pop cx

pop bx

ret

parse\_cmd\_text endp

;

to\_asciiz proc

push ax

push cx

push di

parse\_to\_asciiz:

mov al, ds:[si]

cmp al, ' '

je is\_delimeter

cmp al, 0Dh

je is\_delimeter

cmp al, 09h

je is\_delimeter

cmp al, 0Ah

je is\_delimeter

cmp al, 00h

je is\_delimeter

cmp al, '$'

je is\_delimeter

mov es:[di], al

inc di

inc si

loop parse\_to\_asciiz

is\_delimeter:

mov al, 00h

mov es:[di], al

mov al, '$'

inc di

mov es:[di], al

inc si

pop di

pop cx

pop ax

ret

to\_asciiz endp

strlen proc

push bx

push si

xor ax, ax

start\_strlen:

mov bl, ds:[si]

cmp bl, 00h

je end\_strlen

inc si

inc ax

jmp start\_strlen

end\_strlen:

pop si

pop bx

ret

strlen endp

count\_raws:

xor dx, dx

search:

fread

mov al, [word+2]

mov cl, [buf]

cmp cl, al

je check\_word

jmp search

check\_word:

lea si, word+2

mov al, [si]

mov ah, 1

while:

inc ah

inc si

mov al, [si]

mov cl, [word+1]

cmp ah, cl

jg success

fread

mov cl, [buf]

cmp al, cl

jne search

je while

success:

inc dx

skip:

fread

mov al, 13

mov cl, [buf]

cmp al, cl

jne skip

fread

jmp search

jmp count\_raws\_end

start:

mov ax, @data

mov es, ax

xor ch, ch

mov cl, ds:[80h]

mov cmd\_length, cl

mov si, 82h

lea di, cmd\_text

rep movsb

mov ds, ax

call parse\_cmd\_text

test ax, ax

jne exit

output msg\_prompt\_to\_input

input word

fopen

output msg\_result

jmp count\_raws

count\_raws\_end:

exit:

output msg\_error

pop dx

pop cx

pop ax

mov ax, 4c00h

int 21h

good\_exit:

output crlf

pop dx

pop cx

pop ax

mov ax, dx

puti

mov ax, 4c00h

int 21h

end start

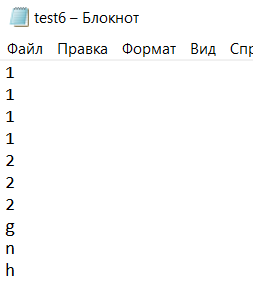


Рисунок 2 – Изначальные вводимые данные

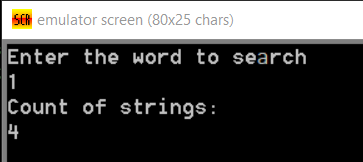


Рисунок 1 – Результат работы программы

**Выводы:** В результате лабораторной работы была выполнена одна задача с использованием основными операциями обработки файлов.