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 №4

на тему:

**«РАБОТА С ФАЙЛАМИ.»**

БГУИР 6-05-0612-02 39

|  |
| --- |
| Выполнил студент группы 353503  ДЕРГУН Николай Олегович |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 4.** Подсчитать число строк с длиной менее указанного значения в файле. Параметры обработки (включая описание путей к файлам) передавать в программу через параметры командной строки.

# 2 Выполнение работы

Для подсчёта числа строк, необходимо обработать параметры командной строки, хранимые в PSP, сделав из них строку с нулём на конце.

Программа принимает строго два параметра, разделённых пробелами. Первый параметр должен являться путём к файлу, а второй – числом, количество строк меньших которого необходимо подсчитать. Также необходимо организовать чтение из файла для этого используются функции DOS, обёрнутые в макросы.

Листинг кода 1 – Основная процедура программы main

main    proc

        data\_to\_ds

        mov ah, 62h

        int 21h

        mov es, bx

        mov al, es:[80h]

        or al, al

        jz \_no\_args

        ; argc parse

        mov cx, -1

        mov di, 81h

        mov bx, es

        mov ds, bx

find\_param:

        mov al, ' '

        repz scasb

        dec di

        push di

        data\_to\_ds

        inc byte ptr argc

        mov ax, es

        mov ds, ax

        mov si, di

scan\_params:

        lodsb

        cmp al, 0Dh

        je params\_ended

        cmp al, 20h

        jne scan\_params

        dec si

        mov byte ptr [si], 0

        mov di, si

        inc di

        jmp short find\_param

params\_ended:

        dec si

        data\_to\_ds

        mov ah, 2

        cmp ah, argc

        je file\_work

        mov ah, 3

        cmp ah, argc

        jne \_wrong\_args\_num

        pop di

        cmp byte ptr es:[di], 0Dh

        jne \_wrong\_args\_num

        jmp file\_work

\_no\_args:

        jmp no\_args

file\_work:

        mov ax, es

        mov ds, ax

        pop si

        call check\_number

        cmp bx, 1

        jne bad\_size

        data\_to\_ds

        mov num\_ptr, si

        pop si

        mov path\_ptr, si

        fopen

        call contract\_func

        or ax, ax

        jz skip\_output

        call print\_result

skip\_output:

        fclose

        jmp main\_ret

\_wrong\_args\_num:

        jmp wrong\_args\_num

file\_err:

        cmp ax, 02h

        je file\_not\_found

        cmp ax, 03h

        je path\_not\_found

        cmp ax, 05h

        je access\_denied

        lea dx, msg\_other\_file\_err

        jmp print\_err

path\_not\_found:

        lea dx, msg\_path\_not\_found

        jmp print\_err

file\_not\_found:

        lea dx, msg\_file\_not\_found

        jmp print\_err

access\_denied:

        lea dx, msg\_access\_denied

        jmp print\_err

bad\_size:

        lea dx, msg\_bad\_size

        jmp print\_err

no\_args:

        lea dx, msg\_no\_args

        jmp print\_err

wrong\_args\_num:

        lea dx, msg\_wrong\_num\_args

print\_err:

        mov ax, @data

        mov ds, ax

        call printl

main\_ret:

        mov ah, 4Ch

        int 21h

main    endp

Для чтения из файла и подсчёта строк, реализована процедура contract\_func.

Листинг кода 2 – Процедура contract\_func

contract\_func proc ; return in ax if reading has finished without error

                   ; 1 - no error, 0 - was error

        push bx

        push cx

        push dx

        push si

        push di

        push bp

        push es

        push ds

        data\_to\_ds

        mov ax, ds

        mov es, ax

        xor si, si

        xor bp, bp ; not ended flag

        mov str\_beg\_ptr, offset buf

fread\_loop:

        push bx

        mov bx, descriptor

        mov ah, 3Fh

        lea dx, buf

        mov cx, buf\_size

        int 21h

        pop bx

        jc \_fread\_error

        or ax, ax

        jz \_manage\_fe

        mov cx, ax

        lea di, buf

        ;check reminder from prev buf line

        or bp, bp

        jnz has\_rem

        mov si, 0

has\_rem:

        mov bp, 0

parse\_buffer:

        mov al, 0Ah

        repne scasb

        dec di

        cmp es:[di], al

        je manage\_nl

manage\_be:

        inc di

        mov bp, 1

        sub di, str\_beg\_ptr

        add si, di

        add di, str\_beg\_ptr

        mov word ptr str\_beg\_ptr, offset buf

        jmp fread\_loop

\_fread\_error:

        jmp fread\_error

\_fread\_loop:

        jmp fread\_loop

\_func\_ret:

        jmp func\_ret

\_manage\_fe:

        jmp manage\_fe

manage\_nl:

        inc di

        sub di, str\_beg\_ptr

        add si, di

        add di, str\_beg\_ptr

        ;dec si ; to remove 0Dh from the end

        sub si, 2

        cmp si, [required\_size]

        jae isnt\_required\_size

        inc word ptr [str\_cntr]

isnt\_required\_size:

        xor si, si

        mov bp, 0

        mov word ptr str\_beg\_ptr, offset buf

        or cx, cx

        jz \_fread\_loop

        mov word ptr str\_beg\_ptr, di

        jmp parse\_buffer

manage\_fe:

        ;check reminder from prev buf line

        or bp, bp

        jnz \_has\_rem

        mov si, 0

\_has\_rem:

        mov bp, 0

        cmp si, [required\_size]

        jae func\_ret

        inc word ptr [str\_cntr]

        jmp func\_ret

fread\_error:

        xor ax, ax

        lea dx, msg\_fread\_err

        call printl

        jmp func\_ret\_only

func\_ret:

        mov ax, 1

func\_ret\_only:

        pop ds

        pop es

        pop bp

        pop di

        pop si

        pop dx

        pop cx

        pop bx

        ret

contract\_func endp

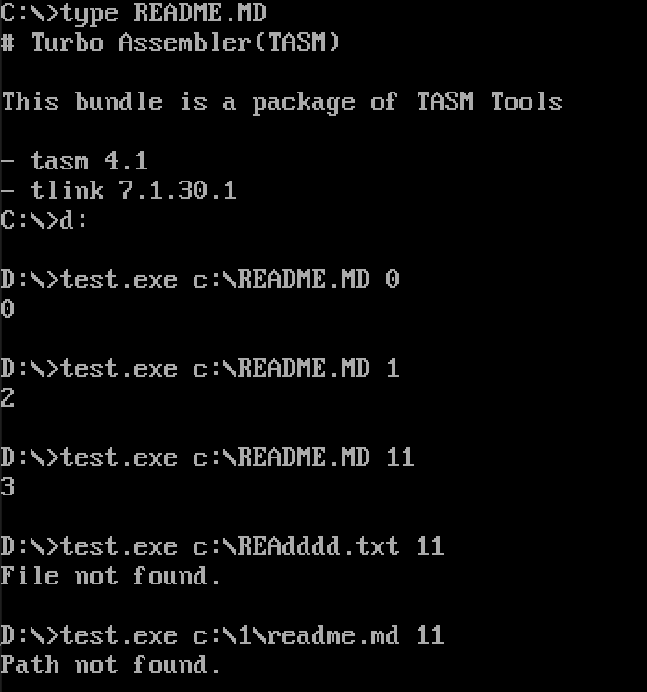


Рисунок 1 – Пример выполнения программы

# Вывод

В ходе выполнения лабораторной работы были получены навыки работы с файлами с использованием функций DOS. Ознакомились с устройством PSP.