

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 253505

Снежко Максим Андреевич

(дата, подпись студента)

Проверил ассистент кафедры
информатики

Романюк Максим Валерьевич

(дата, подпись преподавателя)

Минск 2023

Цель работы: Задание 1. Вариант 21. Удалить в файле строки все информативные (не пустые) строки.

Ход работы: На рисунке 1 представлена передача файла в командную строку. На рисунке 2 представлено содержание строки. На рисунке 3 представлен результат работы в том же файле.

Листинг 1 – Исходный код программы задания 1

```
.model small
.data
bf db 200 dup(0h)
noDataStr db "No data!$"
checkConsole db "Check console file...", 0Ah, 0Dh, "$"
start_algorithm db "Start algorithm", 0Ah, 0Dh, "$"
end_algorithm db "End of algorithm", 0Ah, 0Dh, "$"
error_open_file db "Open file error!", 0Ah, 0Dh, "$"
error_delete_file db "Error delete file!", 0Ah, 0Dh, "$"
error_rename_file db "Error rename file!", 0Ah, 0Dh, "$"
error_create_file db "Error create file!", 0Ah, 0Dh, "$"
error_close_file db "Error close file!", 0Ah, 0Dh, "$"
temp_filename db 'temp.txt', 0
error_read_file db "Error read file!", 0Ah, 0Dh, "$"
buffer db 1h dup(?)
handle_in dw ?
handle_out dw ?
new_handle_in dw ?
index dw 0
end dw 0
okay dw 0
file_name db 15 dup('$')
.code
getComArgs proc
    mov di, 80h
    mov cl, es:[di]
    cmp cl, 1h
    jle empty_cmd
    mov cx, -1h
    mov di, 81h
    mov al, ' '
    repe scasb
    dec di
    mov bx, 0h
get_file_name:
```

```

        cmp es:[di],0Dh
        je set_string_end
        mov al,es:[di]
        mov file_name[bx],al
        inc bx
        inc di
        jmp get_file_name
set_string_end:
        mov file_name[bx], 0h
        ret
empty_cmd:
        lea dx, noDataStr
        mov ah, 09h
        int 21h
        mov ax,4C00h
        int 21h
getComArgs endp
start:
        mov ax, @data
        mov ds, ax
        push dx
        push ax
        lea dx, checkConsole
        mov ah, 09h
        int 21h
        pop ax
        pop dx
        call getComArgs ; check argument
        mov ah,3Dh ; open file
        mov al,0h
        lea dx,file_name
        int 21h
        jc print_open_file_error ; if error (flag C = 1)
        mov bx,ax
        mov handle_in, ax
        mov ah, 3Ch
        mov cx, 0
        lea dx, temp_filename
        int 21h
        mov ah, 3Dh
        mov al, 1h
        lea dx, temp_filename
        int 21h
        jc print_open_file_error
        mov handle_out, ax

```

```

    mov cx, 0h
    push dx
    push ax
    lea dx, start_algorithm
    mov ah, 09h
    int 21h
    pop ax
    pop dx
read_loop:
    cmp end, 1
    je close_files
    mov ah, 3Fh
    push si
    mov si, index
    lea dx, buffer
    mov bx, handle_in
    mov cx, 1
    int 21h
    mov al, [buffer]
    pop si
    mov [bf + si], al
    cmp al, 0    ; end of file
    je end_of_file
    cmp al, 13
    jne proverka_10
    jmp end_proverka
proverka_10:
    cmp al, 10
    je next
proverka_32:
    cmp al, 32
    jne provarka_9
    jmp end_proverka
provarka_9:
    cmp al, 9
    jne not_empty
end_proverka:
    cmp al, 13
    je next_word
next:
    push si
    mov si, index
    inc si
    mov index, si
    pop si

```

```

        inc si
        jmp read_loop
not_empty:
        mov okay, 1
        push si
        mov si, index
        inc si
        mov index, si
        pop si
        inc si
        jmp read_loop
next_word:
        inc si
        mov [bf + si], '$'
        cmp okay, 0
        je print_str
        mov okay, 0
        push si
        mov si, index
        inc si
        mov index, si
        pop si
        mov si, 0
        jmp read_loop
end_file:
        mov end, 1
        jmp print_str
print_str:
        push dx
        push bx
        push ax
        mov al, bf[0]
        cmp al, 10
        je print_without_10    ; for space abc, cret, newl,
tab, cret
        pop ax
        mov ah, 40h
        mov bx, handle_out
        mov cx, si
        lea dx, bf
        int 21h
        pop bx
        pop dx
        mov si, index
        inc si

```

```

    mov index, si
    mov si, 0
    jc close_files
    jmp read_loop
print_without_10:
    pop ax
    mov ah, 40h
    mov bx, handle_out
    mov cx, si
    dec cx
    lea dx, bf+1
    int 21h
    pop bx
    pop dx
    mov si, index
    inc si
    mov index, si
    mov si, 0
    jc close_files
    jmp read_loop
end_of_file:          ; if pered NULL  tab or space(s)
    cmp okay, 0
    je end_file
close_files:
    mov ah, 3Eh
    mov bx, handle_in
    int 21h
    jc close_file_error
print_end_alg:
    push dx
    push ax
    lea dx, end_algorithm
    mov ah, 09h
    int 21h
    pop ax
    pop dx
close_input:  ; work with files
    mov ah, 3Eh
    mov bx, handle_out
    int 21h
    jc close_file_error
    mov ah, 3Ch
    lea dx, file_name
    mov cx, 0
    int 21h

```

```

        jc create_file_error
        mov new_handle_in, ax
        mov ah, 3Eh
        mov bx, new_handle_in
        int 21h
        jc close_file_error
        mov ah, 3Dh
        lea dx, file_name
        mov al, 1
        int 21h
        jc print_open_file_error
        mov ah, 3Dh
        lea dx, temp_filename
        mov al, 0
        int 21h
        jc print_open_file_error
read:
        mov ah, 3Fh
        mov bx, handle_out
        lea dx, buffer
        mov cx, 1
        int 21h
        jc read_file_error
        cmp buffer, 0
        je end_of_filee
        mov ah, 40h
        mov bx, new_handle_in
        lea dx, buffer
        mov cx, 1
        int 21h
        jmp read
end_of_filee:
        mov ah, 3Eh
        mov bx, handle_out
        int 21h
        jc close_file_error
        mov ah, 3Eh
        mov bx, new_handle_in
        int 21h
        jc close_file_error
        xor dx, dx
        mov ah, 41h
        lea dx, temp_filename
        int 21h
        jc delete_file_error

```

```

jmp exit
delete_file_error:
    lea dx, error_delete_file
    mov ah, 9h
    int 21h
    jmp exit
read_file_error:
    lea dx, error_read_file
    mov ah, 9h
    int 21h
    jmp exit
close_file_error:
    lea dx, error_close_file
    mov ah, 9h
    int 21h
    jmp exit
create_file_error:
    lea dx, error_create_file
    mov ah, 9h
    int 21h
    jmp exit
print_open_file_error:
    lea dx, error_open_file
    mov ah, 9h
    int 21h
    jmp exit
exit:
    mov ah, 4Ch
    int 21h
end start

```

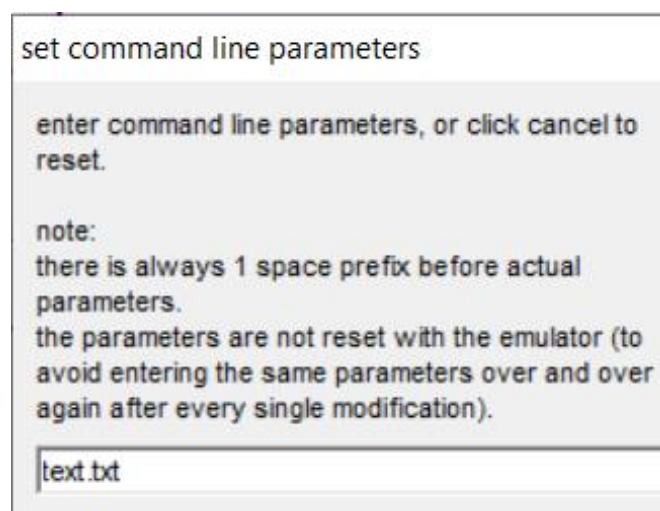


Рисунок 1 – передача файла в командную строку

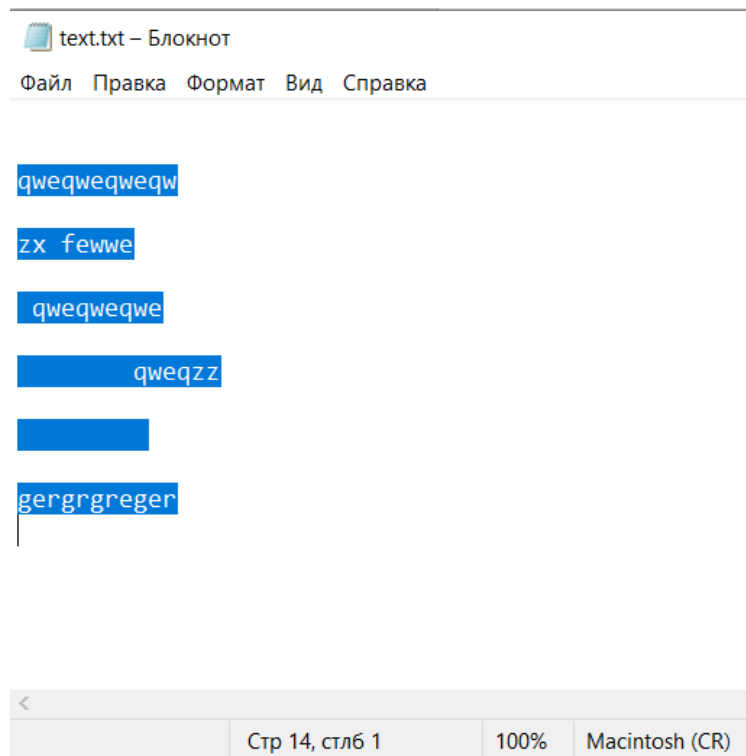


Рисунок 2 – содержание файла

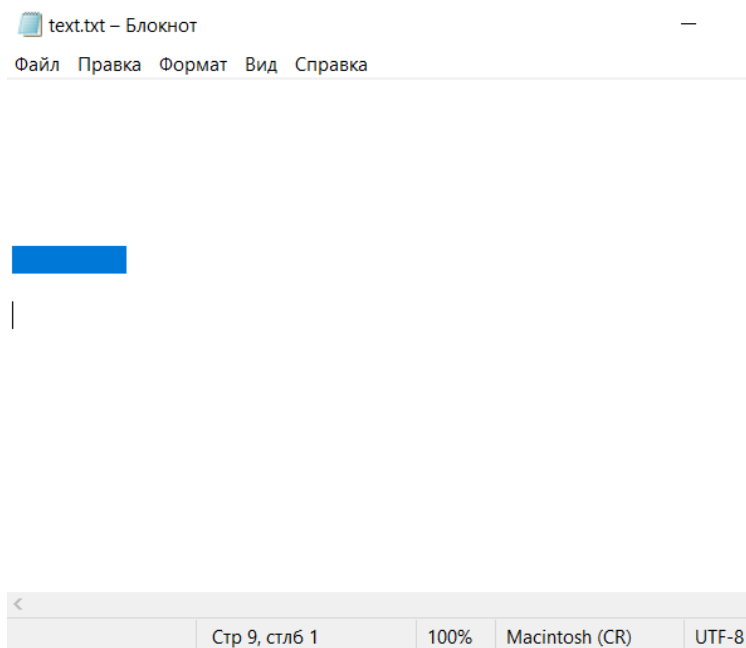


Рисунок 3 – результат работы в том же файле

Вывод: выполнено задание удалить в файле строки все информативные (не пустые) строки. Таким образом были изучение основные команды (прерывания) работы с файлами (создание, открытие, закрытие файлов, работа с данными файлов, поиск файлов, работа с командной строкой, указатель чтения/записи и т. д.) для работы на ассемблере для процессора Intel 8086