



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №10

«Цепочечные команды. Работа с файлами»

ДИСЦИПЛИНА: «Машинно-зависимые языки программирования»

Выполнил: студент гр. ИУК4-31Б _____ (Отрошенко Т. В.)
(Подпись)

Проверил: _____ (Амеличева К. А.)
(Подпись)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2021

Цель работы: реализовать основные операции работы с файлами: открытие файла для чтения/записи, ввод-вывод в файл и т.д.

Задачи:

1. Создать текстовый файл, в котором хранится исходная строка символов, и поместить его в каталог с программой.
2. Составить программу, которая содержит сегменты кода, данных, стека и выполняет:
 - считывание из файла исходной строки.
 - производит над файлом соответствующие варианты действия и выводит результат в другой (новый) созданный файл.
 - при составлении программы использовать цепочечные команды.

Предусмотреть обработку ошибок: файл не найден, ошибка записи, ошибка чтения.

3. Написать вариант программы в формате типа.com, используя работу со строкой как с массивом символов, запустить ее под управлением отладчика, проанализировать содержимое регистров процессора.

Вариант 10

Во второй строке произвольного текстового файла изменить последовательность символов на обратную. Исходную и измененную строку показать на экране. Изменения сохранить в новом файле.

```
.model small

.stack 100h

.386

.data

input_file_name db "ifile.txt", 0
output_file_name db "ofile.txt", 0

input_file_ID dw ?
output_file_ID dw ?

opening_error_message db "File could not be opened", "$"
```

```

creation_error_message db "File could not be created", "$"
read_error_message      db "File could not be read", "$"
write_error_message      db "Error in writing in the file", "$"
completion_message      db "Program completed successfully", "$"

char_buffer db 2, ?
char         db ?, ?
string       db ?

.code

    mPrint macro string
        pusha

        mov ax, 0900h
        lea dx, string
        int 21h

        popa
    endm

start:

    mov ax, @data
    mov ds, ax
    mov es, ax

    mov ax, 3D00h ; открыть
    xor cx, cx
    lea dx, input_file_name
    int 21h

```

```

    mov input_file_ID, ax
    jc opening_error

    mov ax, 3C00h ; создать
    xor cx, cx
    lea dx, output_file_name
    int 21h

    mov output_file_ID, ax
    jc creation_error

; работа с файлом
    lea di, string
    lea si, string

    mov ax, 0
    push ax
    push ax

file_read_loop:
    mov ax, 3F00h
    mov bx, input_file_ID
    mov cx, 1
    lea dx, char
    int 21h

    jc read_error

    cmp ax, 0
    je end_file_read_loop

```

```

    mov al, char
    cmp al, 0Dh
    je end_of_line

    pop bx
    pop cx
    cmp bx, 1
    jne save_char

    inc cx
    xor ah, ah
    push ax
    jmp continue_file_read_loop

save_char:
    stosb

continue_file_read_loop:
    push cx
    push bx
    jmp file_read_loop

end_of_line:
    pop bx
    pop cx
    cmp bx, 1
    jne save_end_line

reverse_loop:

```

```

    pop ax

    stosb

    loop reverse_loop

save_end_line:

    push cx

    push bx

    mov al, 0Dh

    stosb


    mov ax, 3F00h

    mov bx, input_file_ID

    mov cx, 1

    lea dx, char

    int 21h


    pop ax

    inc ax

    push ax


    jmp file_read_loop

end_file_read_loop:

; запись

    sub di, si


    mov ax, 4000h

    mov bx, output_file_ID

    mov cx, di

```

```

    lea dx, string

    int 21h

    jc write_error
; закрытие

    mov ax, 3E00h

    mov bx, input_file_ID

    int 21h

    mov ax, 3E00h

    mov bx, output_file_ID

    int 21h

; успешное завершение

    mPrint completion_message

    jmp end_program

; завершение с ошибками
opening_error:

    mPrint opening_error_message

    jmp end_program

creation_error:

    mPrint creation_error_message

    jmp end_program

read_error:

    mPrint read_error_message

    jmp end_program

```

```

write_error:

    mPrint read_error_message

    jmp end_program

; конец

end_program:

    mov ax, 4C00h

    int 21h

end start

end

```

Результаты выполнения:

Input.txt	Output.txt
tea cup	tea cup
tea cup	puc aet
tea cup	tea cup
tea cup	tea cup
tea cup0	tea cup0
tea cup0	tea cup0
tea cup0	tea cup0

Вывод: в ходе выполнения лабораторной работы были приобретены навыки написания программ с циклами на языке Ассемблер.