|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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. ИУК4-32Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Амеличева К.А. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2021

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

**Вариант 8**

Во второй строке произвольного текстового файла в словах нечетной длины удалить среднюю букву. Исходную и измененную строку показать на экране. Изменения сохранить в новом файле.

**Возможные ошибки:**

Файл не был открыт – проверка флагов после команды 3Dh из int 21h и кода ошибки в AX

Файл не был прочтен – проверка флагов после команды 3Fh из int 21h

Файл не был найден – проверка флагов после команды 3Dh из int 21h и кода ошибки в AX

Файл не был создан – проверка флагов после команды 3Сh из int 21h

Запись в файл не удалась – проверка флагов после команды 40h из int 21h

**Листинг:**

.exe

.model small

.stack 100h

.386

.data

CR = 0Dh

LF = 0Ah

FileName db "sentence.txt0", "$"

FDescr dw 0

NewFile db "newfile.txt0", "$"

FDescrNew dw 0

Buffer dw 0

String dw 256 dup(0)

index dw 0

oldString dw 256 dup(0)

newString dw 256 dup(0)

isFound db 0

isNotLast db 0

MessageError1 db CR, LF, "File was not opened !", "$"

MessageError2 db CR, LF, "File was not read !", "$"

MessageError3 db CR, LF, "File was not founded!", "$"

MessageError4 db CR, LF, "File was not created!", "$"

MessageError5 db CR, LF, "Error in writing in the file!", "$"

MessageEnd db CR, LF, "Program was successfully finished!", "$"

nextLine db CR, LF, '$'

print\_string macro

mov ah, 09h

int 21h

endm

mFormString macro FDescr, Buffer, index, String, oldString, newString, isNotLast

LOCAL M1

LOCAL M2

LOCAL M3

LOCAL CYCLE

LOCAL CYCLE\_END

LOCAL ODD

LOCAL MIDDLE

LOCAL LAST\_MIDDLE

LOCAL LAST\_END

LOCAL NOT\_DEC

push index

mov index, 0

xor si, si

M1:

mov ah, 3fh

mov bx, FDescr

mov cx, 1

mov dx, offset Buffer

int 21h

cmp ax, 0

je M3

mov ax, Buffer

cmp ax, CR

je M2

mov bx, index

mov oldString[bx], ax

inc bx

mov index, bx

jmp M1

M2:

mov isNotLast, 1

mov bx, index

mov oldString[bx], CR

inc index

M3:

mov cx, index

xor bx, bx

xor si, si

lea di, newString

CYCLE:

cmp byte ptr oldString[bx], ' '

je ODD

jmp CYCLE\_END

ODD:

mov ax, bx

sub ax, si

mov dx, 2

div dl

cmp ah, 1

je MIDDLE

push cx

mov cx, bx

sub cx, si

lea dx, oldString

add si, dx

rep movsb

mov byte ptr [di], ' '

inc di

sub si, dx

inc si

pop cx

jmp CYCLE\_END

MIDDLE:

push cx

xor cx, cx

mov cl, al

lea dx, oldString

add si, dx

rep movsb

inc si

mov cl, al

rep movsb

mov byte ptr [di], ' '

inc di

sub si, dx

inc si

pop cx

CYCLE\_END:

inc bx

dec cx

cmp cx, 0

jne CYCLE

mov ax, bx

cmp byte ptr isNotLast, 1

jne NOT\_DEC

dec ax

NOT\_DEC:

sub ax, si

mov dx, 2

div dl

cmp ah, 1

je LAST\_MIDDLE

mov cx, bx

sub cx, si

lea dx, oldString

add si, dx

rep movsb

jmp LAST\_END

LAST\_MIDDLE:

xor cx, cx

mov cl, al

lea dx, oldString

add si, dx

rep movsb

inc si

mov cl, al

rep movsb

LAST\_END:

lea ax, newString

sub di, ax

mov cx, di

mov ax, di

lea di, String

pop index

add di, index

add index, ax

lea si, newString

rep movsb

mov oldString[bx], '$'

mov bx, ax

mov newString[bx], '$'

lea dx, oldString

print\_string

lea dx, nextLine

print\_string

lea dx, newString

print\_string

endm

.code

start:

mov ax, @data

mov ds, ax

mov es, ax

mov ah, 10h

mov al, 3

sub bl, bl

int 10h

mov ah, 6

sub al, al

mov bh, 0F0h

sub cx, cx

mov dx, 184Fh

int 10h

mov ah, 2

xor dx, dx

sub bh, bh

int 10h

mov ah, 3Dh

xor al, al

mov dx, offset FileName

xor cx, cx

int 21h

mov FDescr, ax

jnc M1

jmp Er1

M1:

mov ah, 3ch

xor cx, cx

mov dx, offset NewFile

int 21h

mov FDescrNew, ax

jnc M2

jmp Er3

M2:

mov ah, 3fh

mov bx, FDescr

mov cx, 1

mov dx, offset Buffer

int 21h

jnc M3

jmp Er2

M3:

cmp ax, 0

je M4

mov ax, Buffer

mov bx, index

mov String[bx], ax

inc bx

mov index, bx

cmp isFound, 1

je M2

cmp ax, LF

je MACRO\_CALL

jmp M2

MACRO\_CALL:

mFormString FDescr, Buffer, index, String, oldString, newString, isNotLast

mov isFound, 1

jmp M2

M4:

mov ah, 40h

mov bx, FDescrNew

mov cx, index

mov dx, offset String

int 21h

jnc M5

jmp Er4

M5:

mov ah, 3eh

mov bx, FDescr

int 21h

mov ah, 3eh

mov bx, FDescrNew

int 21h

mov dx, offset MessageEnd

print\_string

jmp Exit

Er1:

cmp ax, 02h

jne M6

lea dx, MessageError3

print\_string

jmp Exit

M6:

lea dx, MessageError1

print\_string

jmp Exit

Er2:

lea dx, MessageError2

print\_string

jmp Exit

Er3:

lea dx, MessageError4

print\_string

jmp Exit

Er4:

lea dx, MessageError5

print\_string

jmp Exit

Exit:

mov ah, 07h

int 21h

mov ax, 4c00h

int 21h

end start

.com

sc segment 'code'

assume cs:sc, ds:sc

.386

org 256

start proc

print\_string macro

mov ah, 09h

int 21h

endm

mFormString macro FDescr, Buffer, index, String, oldString, newString, isNotLast

LOCAL M1

LOCAL M2

LOCAL M3

LOCAL CYCLE

LOCAL CYCLE\_END

LOCAL ODD

LOCAL MIDDLE

LOCAL LAST\_MIDDLE

LOCAL LAST\_END

LOCAL NOT\_DEC

push index

mov index, 0

xor si, si

M1:

mov ah, 3fh

mov bx, FDescr

mov cx, 1

mov dx, offset Buffer

int 21h

cmp ax, 0

je M3

mov ax, Buffer

cmp ax, CR

je M2

mov bx, index

mov oldString[bx], ax

inc bx

mov index, bx

jmp M1

M2:

mov isNotLast, 1

mov bx, index

mov oldString[bx], CR

inc index

M3:

mov cx, index

xor bx, bx

xor si, si

lea di, newString

CYCLE:

cmp byte ptr oldString[bx], ' '

je ODD

jmp CYCLE\_END

ODD:

mov ax, bx

sub ax, si

mov dx, 2

div dl

cmp ah, 1

je MIDDLE

push cx

mov cx, bx

sub cx, si

lea dx, oldString

add si, dx

rep movsb

mov byte ptr [di], ' '

inc di

sub si, dx

inc si

pop cx

jmp CYCLE\_END

MIDDLE:

push cx

xor cx, cx

mov cl, al

lea dx, oldString

add si, dx

rep movsb

inc si

mov cl, al

rep movsb

mov byte ptr [di], ' '

inc di

sub si, dx

inc si

pop cx

CYCLE\_END:

inc bx

dec cx

cmp cx, 0

jne CYCLE

mov ax, bx

cmp byte ptr isNotLast, 1

jne NOT\_DEC

dec ax

NOT\_DEC:

sub ax, si

mov dx, 2

div dl

cmp ah, 1

je LAST\_MIDDLE

mov cx, bx

sub cx, si

lea dx, oldString

add si, dx

rep movsb

jmp LAST\_END

LAST\_MIDDLE:

xor cx, cx

mov cl, al

lea dx, oldString

add si, dx

rep movsb

inc si

mov cl, al

rep movsb

LAST\_END:

lea ax, newString

sub di, ax

mov cx, di

mov ax, di

lea di, String

pop index

add di, index

add index, ax

lea si, newString

rep movsb

mov oldString[bx], '$'

mov bx, ax

mov newString[bx], '$'

lea dx, oldString

print\_string

lea dx, nextLine

print\_string

lea dx, newString

print\_string

endm

mov ah, 10h

mov al, 3

sub bl, bl

int 10h

mov ah, 6

sub al, al

mov bh, 0F0h

sub cx, cx

mov dx, 184Fh

int 10h

mov ah, 2

xor dx, dx

sub bh, bh

int 10h

mov ah, 3Dh

xor al, al

mov dx, offset FileName

xor cx, cx

int 21h

mov FDescr, ax

jnc M1

jmp Er1

M1:

mov ah, 3ch

xor cx, cx

mov dx, offset NewFile

int 21h

mov FDescrNew, ax

jnc M2

jmp Er3

M2:

mov ah, 3fh

mov bx, FDescr

mov cx, 1

mov dx, offset Buffer

int 21h

jnc M3

jmp Er2

M3:

cmp ax, 0

je M4

mov ax, Buffer

mov bx, index

mov String[bx], ax

inc bx

mov index, bx

cmp isFound, 1

je M2

cmp ax, LF

je MACRO\_CALL

jmp M2

MACRO\_CALL:

mFormString FDescr, Buffer, index, String, oldString, newString, isNotLast

mov isFound, 1

jmp M2

M4:

mov ah, 40h

mov bx, FDescrNew

mov cx, index

mov dx, offset String

int 21h

jnc M5

jmp Er4

M5:

mov ah, 3eh

mov bx, FDescr

int 21h

mov ah, 3eh

mov bx, FDescrNew

int 21h

mov dx, offset MessageEnd

print\_string

jmp Exit

Er1:

cmp ax, 02h

jne M6

lea dx, MessageError3

print\_string

jmp Exit

M6:

lea dx, MessageError1

print\_string

jmp Exit

Er2:

lea dx, MessageError2

print\_string

jmp Exit

Er3:

lea dx, MessageError4

print\_string

jmp Exit

Er4:

lea dx, MessageError5

print\_string

jmp Exit

Exit:

mov ah, 07h

int 21h

mov ax, 4c00h

int 21h

CR = 0Dh

LF = 0Ah

FileName db "sentence.txt0", "$"

FDescr dw 0

NewFile db "newfile.txt0", "$"

FDescrNew dw 0

Buffer dw 0

String dw 256 dup(0)

index dw 0

oldString dw 256 dup(0)

newString dw 256 dup(0)

isFound db 0

isNotLast db 0

MessageError1 db CR, LF, "File was not opened !", "$"

MessageError2 db CR, LF, "File was not read !", "$"

MessageError3 db CR, LF, "File was not founded!", "$"

MessageError4 db CR, LF, "File was not created!", "$"

MessageError5 db CR, LF, "Error in writing in the file!", "$"

MessageEnd db CR, LF, "Program was successfully finished!", "$"

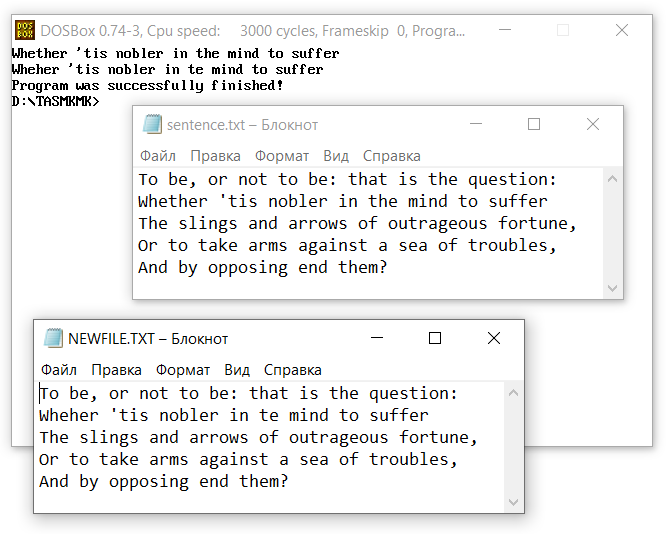
nextLine db CR, LF, '$'

start endp

sc ends

end start

**Результаты работы:**



**Рисунок 1.** Результат

**Вывод:** в ходе выполнения лабораторной работы были получены навыки использования цепочечных команд работы с файлами, создания файла, чтения из файла, записи в файл, обработки ошибок.