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

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | "Фундаментальные науки" |
| КАФЕДРА | "Программное обеспечение ЭВМ, информационные технологии и прикладная математика" |

##### 

О Т Ч Е Т

ЛАБОРАТОРНАЯ РАБОТА №11

|  |  |  |
| --- | --- | --- |
| ДИСЦИПЛИНА: | | "Машинозависимые языки программирования" |
| ТЕМА: | **«**Цепочечные команды. Работа с файлами**»** | |

|  |  |
| --- | --- |
| Выполнил: студент гр. ИТД.Б-32 | Лысенков А.Е.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Проверил: | Амеличева К.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |  |
| --- | --- | --- |
| Дата сдачи (защиты) лабораторной работы: | |  |
| Результаты сдачи (защиты):  Количество рейтинговых баллов |  | |
| Оценка |  | |

Калуга, 2015г.

## Цель работы

Научиться использовать цепочечные команды.

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

## Постановка 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 варианту действия и выводит результат в другой (новый) созданный файл. При составлении программы использовать цепочечные команды. Отлаженную программу дополнить комментарием, описывающим ее назначение, и сохранить в файле lab.asm для отчета. Предусмотреть обработку ошибок: файл ненайден, ошибка записи, ошибка чтения.
3. Написать вариант программы в формате типа .com, используя работу со строкой как с массивом символов. Отлаженную программу сохранить в файле lab1.asm для отчета.

**Задание:**

**Вариант 11**

Определить количество слов наибольшей длины

**Код программы:**

.model small

.stack 100h

assume cs: code, ds: data

data segment

minus db '-$'

color db 3Eh

file db 'data.txt', 0

file2 db 'data2.txt', 0

link dw 0d

link2 dw 0d

content db 62, 0, 62 dup ('$')

content2 db 62, 0, 62 dup ('$')

maxLength db 0d

count db 0d

msgInput db ' > $'

msgNext db ' --- Press <Enter> ---$'

msgErrLocate db'Error:Can't locate$'

msgErrCreate db 'Error:Can'tcreate$'

msgErrRead db 'Error:Can't read $'

msgErrWrite db 'Error:Can't write$'

msgMaxLength db ' Max length: $'

msgCount db ' Max length count: $'

data ends

code segment

mBr macro

mov ax, 0200h

mov dl, 0Ah

int 21h

endm

mPrintStr macro string

mov ax, 0900h

mov dx, offset string

int 21h

endm

mError macro msg

mPrintStr msgInput

mPrintStr msg

jmp exit

endm

mOpenFile macro filename, link

local success

xor cx, cx

mov ax, 3D00h

mov dx, offset filename

int 21h

mov link, ax

jnc success

mError msgErrLocate

success:

endm

mCreateFile macro filename: REQ, link: REQ

local success

xor cx, cx

mov dx, offset filename

mov ax, 3C00h

int 21h

mov link, ax

jnc success

mError msgErrCreate

success:

endm

mLengthString macro string: REQ

local nextSymbol

push si

push bx

xor ax, ax

mov si, offset string

nextSymbol:

inc si

inc ax

mov bl, [si]

cmp bl, '$'

jne nextSymbol

pop bx

pop si

endm

mReadFile macro link:REQ,content:REQ

local success

mov cx, 62d

mov bx, link

mov dx, offset content

mov ax, 3F00h

int 21h

jnc success

mError msgErrRead

success:

endm

mWriteFile macro link:REQ,content

local success

mov bx, link

xor cx, cx

mLengthString content

mov cx, ax

mov dx, offset content

mov ax, 4000h

int 21h

jnc success

mError msgErrWrite

success:

endm

mCloseFile macro link

mov bx, link

mov ax, 3E00h

int 21h

endm

mLength macro link: REQ

local nextSymbol, done

push cx

push dx

xor ax, ax

mov cx, 62d

nextSymbol:

lodsb

cmp al, ' '

je done

cmp al, '$'

je done

loop nextSymbol

done:

mov ax, 62d

sub ax, cx

inc ax

pop dx

pop cx

endm

mFindMaxLength macro string: REQ

local nextWord, less, addThat

mov si, offset string

inc si

mov cl, 0d

mov count, 0d

nextWord:

xor ax, ax

mov bx, si

mLength bx

cmp al, cl

je addThat

jl less

mov cl, al

pop dx

mov count, 0

addThat:

add count, 1

less:

lodsb

cmp al, '$'

jne nextWord

mov maxLength, cl

endm

mWriteWord macro

local nextSymbol, done

push cx

xor ax, ax

mov cx, 62d

cld

nextSymbol:

lodsb

cmp al, ' '

je done

cmp al, '$'

je done

stosb

loop nextSymbol

done:

stosb

pop cx

endm

mWriteTask macro

local nextWord, skip

mov si, offset content

;mov di, offset content2

inc si

nextWord:

mov bx, si

mLength bx

cmp al, curLength

jne skip

push si

mov si, cx

mWriteWord

pop si

mov cx, si

skip:

inc si

jne nextWord

mov count, cl

endm

mStringNumb macro

local nextDigit2, printDigit

cld

xor cx, cx

mov bx, 10d

nextDigit2:

xor dx, dx

div bx

push dx

inc cx

cmp ax, 0d

jg nextDigit2

xor ax, ax

mov ax, 0200h

printDigit:

pop ax

add ax, '0'

stosb

loop printDigit

xor ax, ax

mov al, ' '

stosb

endm

start:

mov ax, data

mov ds, ax

mov es, ax

mOpenFile file, link

mCreateFile file2, link2

mReadFile link, content

mFindMaxLength content

mBr

mPrintStr msgCount

xor dx, dx

mov dl, count

add dl, '0'

mov ah, 02h

int 21h

mBr

mPrintStr msgMaxLength

xor dx, dx

mov dl, maxLength

add dl, '0'

mov ah, 02h

int 21h

mBr

xor ax, ax

mov al, count

mov di, offset content2

mStringNumb

mWriteFile link2, content2

exit:

mCloseFile link

mCloseFile link2

mPrintStr msgNext

mov ax, 0A00h

lea dx, content

int 21h

mov ax, 0700h

int 12h

mov ax, 4C00h

int 21h

code ends

end start

**.com формат:**

sc segment 'code'

assume cs: sc, ds: sc

org 256

start proc

mBr macro

mov ax, 0200h

mov dl, 0Ah

int 21h

endm

mPrintStr macro string

mov ax, 0900h

mov dx, offset string

int 21h

endm

mError macro msg

mPrintStr msgInput

mPrintStr msg

jmp exit

endm

mOpenFile macro filename, link

local success

xor cx, cx

mov ax, 3D00h

mov dx, offset filename

int 21h

mov link, ax

jnc success

mError msgErrLocate

success:

endm

mCreateFile macro filename: REQ, link: REQ

local success

xor cx, cx

mov dx, offset filename

mov ax, 3C00h

int 21h

mov link, ax

jnc success

mError msgErrCreate

success:

endm

mLengthString macro string: REQ

local nextSymbol

push si

push bx

xor ax, ax

mov si, offset string

nextSymbol:

inc si

inc ax

mov bl, [si]

cmp bl, '$'

jne nextSymbol

pop bx

pop si

endm

mReadFile macro link: REQ, content: REQ

local success

mov cx, 62d

mov bx, link

mov dx, offset content

mov ax, 3F00h

int 21h

jnc success

mError msgErrRead

success:

endm

mWriteFile macro link: REQ, content

local success

mov bx, link

xor cx, cx

mLengthString content

mov cx, ax

mov dx, offset content

mov ax, 4000h

int 21h

jnc success

mError msgErrWrite

success:

endm

mCloseFile macro link

mov bx, link

mov ax, 3E00h

int 21h

endm

mLength macro link: REQ

local nextSymbol, done

push cx

push dx

xor ax, ax

mov cx, 62d

nextSymbol:

lodsb

cmp al, ' '

je done

cmp al, '$'

je done

loop nextSymbol

done:

mov ax, 62d

sub ax, cx

inc ax

pop dx

pop cx

endm

mFindMaxLength macro string: REQ

local nextWord, less, addThat

mov si, offset string

inc si

mov cl, 0d

mov count, 0d

nextWord:

xor ax, ax

mov bx, si

mLength bx

cmp al, cl

je addThat

jl less

mov cl, al

pop dx

mov count, 0

addThat:

add count, 1

less:

lodsb

cmp al, '$'

jne nextWord

mov maxLength, cl

endm

mWriteWord macro

local nextSymbol, done

push cx

xor ax, ax

mov cx, 62d

cld

nextSymbol:

lodsb

cmp al, ' '

je done

cmp al, '$'

je done

stosb

loop nextSymbol

done:

stosb

pop cx

endm

mWriteTask macro

local nextWord, skip

mov si, offset content

;mov di, offset content2

inc si

nextWord:

mov bx, si

mLength bx

cmp al, curLength

jne skip

push si

mov si, cx

mWriteWord

pop si

mov cx, si

skip:

inc si

jne nextWord

mov count, cl

endm

mStringNumb macro

local nextDigit2, printDigit

cld

xor cx, cx

mov bx, 10d

nextDigit2:

xor dx, dx

div bx

push dx

inc cx

cmp ax, 0d

jg nextDigit2

xor ax, ax

mov ax, 0200h

printDigit:

pop ax

add ax, '0'

stosb

loop printDigit

xor ax, ax

mov al, ' '

stosb

endm

mOpenFile file, link

mCreateFile file2, link2

mReadFile link, content

mFindMaxLength content

mBr

mPrintStr msgCount

xor dx, dx

mov dl, count

add dl, '0'

mov ah, 02h

int 21h

mBr

mPrintStr msgMaxLength

xor dx, dx

mov dl, maxLength

add dl, '0'

mov ah, 02h

int 21h

mBr

xor ax, ax

mov al, count

mov di, offset content2

mStringNumb

mWriteFile link2, content2

exit:

mCloseFile link

mCloseFile link2

mPrintStr msgNext

mov ax, 0A00h

lea dx, content

int 21h

mov ax, 0700h

int 12h

mov ax, 4C00h

int 21h

minus db '-$'

color db 3Eh

file db 'data.txt', 0

file2 db 'data2.txt', 0

link dw 0d

link2 dw 0d

content db 62, 0, 62 dup ('$')

content2 db 62, 0, 62 dup ('$')

maxLength db 0d

count db 0d

msgInput db ' > $'

msgNext db ' --- Press <Enter> ---$'

msgErrLocate db'Error:Can't locate$'

msgErrCreate db'Error:Can't create$'

msgErrRead db'Error:Can't read$'

msgErrWrite db 'Error:Can't write$'

msgMaxLength db ' Max length: $'

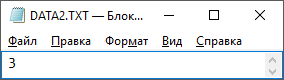
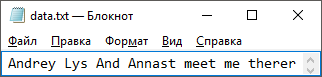
msgCount db ' Max length count: $'

start endp

sc ends

end start

**Результат работы программы:**



# Вывод:

В ходе лабораторной работы были получены навыки по использованию цепочечных команд и реализации основных операций работы с файлами: открытие файла для чтения/записи, ввод-вывод в файл и т.д.