МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

# Кафедра электронных вычислительных машин

Отчет по предмету

Программирование на языке ассемблера

Лабораторная работа №5

**«Работа с файлами.»**

|  |  |
| --- | --- |
| **Выполнил:**  Студент группы 230501  Кочеров Р.С. | **Проверил:**  Туровец Н.О. |

Минск 2023

Цель работы: Ознакомиться с основными операциями обработки файлов, получить понятие о работе с параметрами командной строки.

**Теоретические сведения:**

1. Работа с файлами. Для работы с файлами в данной лабораторной работе лучше всего использовать функции DOS, которые обращаются к файлу через 16-битный идентификатор (дескриптор) файла. Такой подход более прост, чем использование более старых описателей файла (37-байтного блока управления файлом FCB) или функций низкого уровня доступа к диску (прерывание BIOS 13h). Первые пять значений идентификаторов такого формата инициализируются системой следующим образом: -- 0 – STDIN – стандартное устройство ввода (клавиатура), -- 1 – STDOUT – стандартное устройство вывода (экран), -- 2 – STDERR – устройство вывода сообщений об ошибках (всегда экран), 39 -- 3 – AUX – последовательный порт (СОМ1), -- 4 – PRN – параллельный порт (LPT1). Работа с файлами выполняется через функции DOS в стандартном порядке: 1) создание или открытие существующего файла; 2) выполнение файловых операций чтения или записи данных; 3) закрытие файла. Дополнительно доступны операции: удаление, поиск и управление. Для создания или открытия существующего файла рекомендуется использовать функции: -- Функция DOS 3Ch (INT 21h) – создать файл: Ввод: AH = 3Ch, СХ = атрибут файла (биты): -- 7 – файл можно открывать разным процессам, -- 6 – не используется, -- 5 – архивный бит (1, если файл не сохранялся), -- 4 – каталог (должен быть 0 для функции 3Ch), -- 3 – метка тома (игнорируется функцией 3Ch), -- 2 – системный файл, -- 1 – скрытый файл, -- 0 – файл только для чтения. DS:DX = адрес ASCIZ-строки с полным именем файла. Вывод: Если CF = 1, то ошибка операции (в AX код ошибки): -- 03h – путь не найден, -- 04h – слишком много открытых файлов, -- 05h – доступ запрещен. Если CF = 0, то операция выполнена успешно: АХ = идентификатор файла. Особенности: Если файл уже существует, функция все равно открывает его, присваивая ему нулевую длину. -- Функция DOS 5Bh (INT 21h) – создать и открыть файл: Ввод: AH = 5Bh, СХ = атрибут файла, DS:DX = адрес ASCIZ-строки с полным именем файла. Вывод: Если CF = 1, то ошибка операции (в AX код ошибки): -- 03h – путь не найден, -- 04h – слишком много открытых файлов, -- 05h – доступ запрещен, -- 50h – файл уже существует. Если CF = 0, то операция выполнена успешно: 40 АХ = идентификатор файла, открытого для чтения/записи в режиме совместимости. -- Функция DOS 5Ah (INT 21h) – создать и открыть временный файл: Ввод: AH = 5Ah, СХ = атрибут файла, DS:DX = адрес ASCIZ-строки с путем, оканчивающимся символом «\», и тринадцатью нулевыми байтами в конце. Вывод: Если CF = 1, то ошибка операции (в AX код ошибки): -- 03h – путь не найден, -- 04h – слишком много открытых файлов, -- 05h – доступ запрещен. Если CF = 0, то операция выполнена успешно: АХ = идентификатор файла, открытого для чтения/записи в режиме совместимости (при этом в строку по адресу DS:DX дописывается имя созданного файла). Особенности: Функция создает файл с уникальным именем, который не является на самом деле временным, его следует специально удалять, для чего его имя и записывается в строку в DS:DX. -- Функция DOS 3Dh (INT 21h) – открыть существующий файл: Ввод: AH = 3Dh, AL = режим доступа (биты): -- 0 – открыть для чтения, -- 1 – открыть для записи, -- 3 – 2 – 00 (резерв), -- 6 – 4 – режим доступа для других процессов: 000 – режим совместимости, 001 – все операции запрещены, 010 – запись запрещена 011 – чтение запрещено 100 - запрещений нет. -- 7 – файл не наследуется порождаемыми процессами. DS:DX = адрес ASCIZ-строки с полным именем файла. CL = маска атрибутов файла. Вывод: Если CF = 1, то ошибка операции (в AX код ошибки): -- 02h – файл не найден, -- 03h – путь не найден, -- 04h – слишком много открытых файлов, -- 05h – доступ запрещен, -- 0Ch – неверный режим доступа. Если CF = 0, то операция выполнена успешно: 41 АХ = идентификатор файла. Во всех случаях имя файла (если диск или путь отсутствуют в описании, то системой используются их текущие значения) описывается ASCIZ-строкой (строкой ASCII-символов, оканчивающейся нулем), которая, например, имеет следующий вид: filename db 'с:\data\filename.txt',0

**Текст программы:**

.model tiny

.code

org 100h

Begin:

PUSHA

print welcome

call get\_file\_name

opening\_file:

call open\_file

read\_data:

mov cx,400

mov dx, offset buffer

mov ah, 3Fh

int 21h

CMP ax, 400

je not\_close

inc flag\_close

not\_close:

push cx

mov cx, ax

push ax

push si

mov si, offset buffer

loooop:

mov ax, [si]

CMP al, 10

JE stra

JMP not\_str

stra:

inc count\_length

not\_str:

inc si

loop loooop

pop si

pop ax

pop cx

CMP flag\_close , 1

je close\_file

jmp short read\_data

close\_file:

inc count\_length

mov ax, count\_length

mov number, ax

call output\_int

mov ah, 3Eh

int 21h

int 20h

exit:

CMP ax, 02h

JNE next1

mov ah,9

mov dx,offset new\_line

int 21h

mov ah,9

mov dx,offset error1

int 21h

next1:

CMP ax,03h

JNE next2

mov ah,9

mov dx,offset new\_line

int 21h

mov ah,9

mov dx,offset error2

int 21h

next2:

CMP ax,04h

JNE next3

mov ah,9

mov dx,offset new\_line

int 21h

mov ah,9

mov dx,offset error3

int 21h

next3:

CMP ax,05h

JNE next4

mov ah,9

mov dx,offset new\_line

int 21h

mov ah,9

mov dx,offset error4

int 21h

next4:

CMP ax,0Ch

JNE end\_exit

mov ah,9

mov dx,offset new\_line

int 21h

mov ah,9

mov dx,offset error5

int 21h

end\_exit:

popa

int 20h

;//////////////////////////////////////////PRINT///////////////////////////

macro print str

mov ah,9

mov dx,offset new\_line

int 21h

mov ah,9

mov dx,offset str

int 21h

mov ah,9

mov dx,offset new\_line

int 21h

endm

;//////////////////////////////////////////GET\_FILE///////////////////////////

get\_file\_name proc

xor cx, cx

mov cl, es:[80h] ;this adress contains size of cmd

cmp cl, 0

je empty\_cmd

mov di, 82h ;beginning of cmd

lea si, file\_name ;start of address

get\_symbols:

mov al, es:[di]

cmp al, 0Dh ;compare with end

je opening\_file

cmp al, ' '

je too\_many\_args

mov [si], al;ds:[si]=es:[di] file name=param from cmd

inc di

inc si

jmp get\_symbols

empty\_cmd:

print empty\_cmd\_message

jmp exit

too\_many\_args:

print too\_many\_message

jmp exit

mov [si], 0

ret

get\_file\_name endp

;//////////////////////////////////////////OPEN\_FILE///////////////////////////

open\_file proc

mov dx, offset file\_name ;address ASCIZ-string with name

mov ah, 3Dh ;DOS 3Dh

mov al, 00h ;00 - only reading 0 000 00 0

int 21h ;open file

jc exit ;if error - exit

mov bx, ax ;bx - identefy file

;mov di, 01 ;di identify STDOUT (screen)

print open\_message

ret

open\_file endp

;//////////////////////////////////////////OUTPUT\_INT///////////////////////////

output\_int proc near USES cx,dx,ax

pusha

mov sign\_number,0

CMP number, 0

JE number\_zero

JL out\_negative ;number <0

JMP out\_NOT\_negative ;number >0

out\_negative:

mov sign\_number,1 ;number negative

mov ax, number

NEG ax

mov number, ax

out\_NOT\_negative:

xor ax,ax

mov cx, 5

xor si,si

xor dx,dx

xor bx,bx

CMP sign\_number,1

JE out\_neg

JMP output

out\_neg:

mov symbol\_out , '-'

xor ax,ax

mov ah,9

mov dx,offset symbol\_out

int 21h

output:

xor dx,dx

xor ax,ax

CMP number, 0

JE end\_output

mov AX,number

mov del, 10

IDIV del

mov symbol\_out , dl

mov number, ax

add symbol\_out, 30h

xor bx,bx

mov bx, offset output\_mass

dec bx ;sub bx

xor ax,ax

mov al,symbol\_out

mov si, cx

mov [bx][si] ,al

loop output

JMP end\_output

number\_zero:

mov symbol\_out, '0'

xor ax,ax

mov ah,9

mov dx,offset symbol\_out

int 21h

mov ah,9

mov dx,offset new\_number

int 21h

JMP END

end\_output:

xor bx,bx

mov bx, offset output\_mass

mov [bx+5] ,'$'

xor ax,ax

mov ah,9

mov dx,offset output\_mass

add dx, cx ;offset due to the number of elements(0 in begin str)

int 21h

mov ah,9

mov dx,offset new\_number

int 21h

END:

popa

ret

output\_int endp

welcome db 'Hello! Program start!$'

buffer db 401 DUP ('$')

file\_name\_size equ 70

file\_name db file\_name\_size dup('$')

argc dw 0

error1 db 'ERROR1:FILE NOT FOUND$'

error2 db 'ERROR2:PATH NOT FOUND$'

error3 db 'ERROR3:TOO MANY OPEN FILE$'

error4 db 'ERROR4:ACCESS IS DENIED$'

error5 db 'ERROR5:INVALID ACCESS MODE$'

open\_message db 'File was opened$'

empty\_cmd\_message db 'cmd is empty...Nothing to handle$'

too\_many\_message db 'Too many args were entered$'

new\_line db 13,10,'$'

count dw 0

number dw 0

count\_length dw 0

sign\_number dw 0

del dw 1

buf dw 0

flag\_close db 0

new\_number db 9,32, '$'

str db 'strings in file!'

end\_program db 'Program end!$'

output\_mass db '0', '0', '0', '0', '0', '$', '$'

symbol\_out db ?,'$'

end Begin

**Тест программы:**

**Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание**