

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра электронных вычислительных машин

Отчет по предмету
Проектирование на языке ассемблера
Вариант 1

Лабораторная работа №5
«Работа с файлами»

Выполнил студент группы 150503:
Проверил:

Шарай П.Ю.
Туровец Н.О.

Минск 2022

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.

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),
- 3 – AUX – последовательный порт (COM1),
- 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,

DS:DX = адрес ASCIZ-строки с полным именем 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,

CX = атрибут файла,

DS:DX = адрес ASCIZ-строки с полным именем файла.

-- Функция DOS 5Ah (INT 21h) – создать и открыть временный файл:

Ввод: AH = 5Ah,

CX = атрибут файла,

DS:DX = адрес ASCIZ-строки с путем, оканчивающимся символом «\», и тринадцатью нулевыми байтами в конце.

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):

DS:DX = адрес ASCIZ-строки с полным именем файла.

CL = маска атрибутов файла.

AX = 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 'c:\data\filename.txt',0
```

Для выполнения файловых операций чтения или записи данных рекомендуется использовать функции:

-- Функция DOS 3Fh (INT 21h) – чтение из файла или устройства:

Ввод: AH = 3Fh,

BX = идентификатор файла,

CX = число байт для чтения,

DS:DX = адрес буфера для приема данных.

AX = число считанных байт.

AX меньше, чем заказанное число в CX, то при чтении был достигнут конец файла.

-- Функция DOS 42h (INT 21h) – переместить указатель чтения/записи:

Ввод: AH = 42h,

BX = идентификатор файла,

CX:DX = расстояние, на которое надо переместить указатель

CX:DX = новое значение указателя (в байтах от начала файла).

Эта функция также часто используется для определения длины файла: достаточно вызвать ее с CX = 0, DX = 0, AL = 2, и в

CX:DX будет возвращена длина файла в байтах.

-- Функция DOS 40h (INT 21h) – запись в файл или устройство:

Ввод: AH = 40h,

BX = идентификатор файла,

CX = число байт для записи,

DS:DX = адрес буфера с данными.

AX = число записанных байт.

-- Функция DOS 68h (INT 21h) – сброс файловых буферов DOS на диск:

Ввод: AH = 68h,

BX = идентификатор файла.

-- Функция DOS 0Dh (INT 21h) – сброс всех файловых буферов на диск:

Ввод: AH = 0Dh

Для закрытия файла рекомендуется использовать следующую функцию:

-- Функция DOS 3Eh (INT 21h) – закрыть файл:

Ввод: AH = 3Eh,

BX = идентификатор файла.

Код программы

.model small

.data

Filename db "C:\emu8086\vdribe\C\File.txt", 0;filename

errmes db "Error",'\$' ;error message

Buffer db "\$\$\$\$\$" ;buffer for reading

str_number_mes db "Number : " , '\$' ;output message

FHndl dw 0 ;descriptor

str_number dw 0

.code

start:

mov ax, @data

mov ds, ax

mov es, ax

mov ah, 3dh ;open existing file

mov al, 0 ;for reading

lea dx, Filename ;filename to dx

int 21h

jc error ;CF == 0 - error

mov FHndl, ax ;descriptor

LP:

mov ah, 3Fh ;read from file

lea dx, Buffer ;buffer for reading

mov cx, 1

mov bx, FHndl ;descriptor to bx

int 21h

jc error ;CF == 0 - error

cmp ax, cx ;if num of readed bytes != 1 - end of file

```

jne EOF
push dx                ;save dx
mov dl, Buffer
cmp dl, 10             ;if Buffer == '\n' - one more string
je one_more_str
LP2:
mov ah, 02h           ;output symbol
int 21h
pop dx                ;restore dx
jmp LP
EOF:                   ;end of file
mov bx, FHndl         ;descriptor
mov ah, 3Eh           ;close file
int 21h
inc str_number        ;if end of file - one more string
jnc quit
one_more_str:
inc str_number
jmp LP2
error:                 ;out error message
mov ah, 9h
mov dx, OFFSET errmes
int 21h
quit:
mov ah, 2             ;translation to a new line
mov dx, 10
int 21h
mov ah, 09h          ;output string message
mov dx, offset str_number_mes
int 21h

```

output:

```
    lea di, str_number    ;result for writing
    mov bx, 10            ;divisor
    xor cx,cx             ;cx = 0
    mov ax , str_number
```

ASCII:

```
    xor dx,dx             ;dx = 0
    div bx                ;divide on 10
    add dl,'0'            ;number to symbol
    push dx               ;save dx to stack
    inc cx
    test ax,ax            ;(ax == 0)?
    jnz ASCII
```

recordRezult: ;pop symbols from stack

```
    pop dx               ;pop symbol to dx
    mov ah , 2           ;output symbol
    int 21h
    loop recordRezult
```

mov ax, 4c00h ;end of program

int 21h

end start

Результат работы программы: