**Л/р. № 3. Работа с файлами**

**1. Cоздание нового файла**

%macro task1 1

mov ah, 3Ch ;Функция DOS 3Ch (создание файла)

mov dx, %1 ;Указатель на начало строки в памяти (имя файла)

mov cx,0 ;Нет атрибутов - обычный файл

call createfile;

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

task1 file\_name

;---[ Стандартное завершение программы ]----------------------—

mov AX, 4C00h

int 21h

; Процедура вывода сообщения об ошибке

error\_msg:

mov ah,9

mov dx,s\_error

int 21h ;Вывод сообщения об ошибке

ret

;==================================================================

createfile:

mov ah,3Ch ;Функция DOS 3Ch (создание файла)

mov dx,file\_name ;Имя файла

mov cx,0 ;Нет атрибутов - обычный файл

int 21h ;Обращение к функции DOS

jnc NO\_ERR1 ;Если нет ошибки, то продолжаем

call error\_msg ;Иначе вывод сообщения об ошибке

jmp exit1 ;Выход из программы

NO\_ERR1:

mov [handle],ax ;Сохранение дескриптора файла

close\_file1:

mov ah,3Eh ;Функция DOS 3Eh (закрытие файла)

mov bx,[handle] ;Дескриптор

int 21h ;Обращение к функции DOS

jnc exit1 ;Если нет ошибки, то выход из программы

call error\_msg ;Вывод сообщения об ошибке

exit1:

mov ah,9

mov dx,s\_pak

int 21h ;Вывод строки 'All is ok!'

RET

;==================================================================

;===[ Начало сегмента данных ]==========================================

file\_name db 'new\_file.txt',0

message db 'HELLO',13,10,'world!'

size db 13

s\_error db 'Error!',13,10,'$'

s\_pak db 'All is ok!$'

endline db 13,10,'$'

handle dw 1 ;Дескриптор файла

**2. Перезапись файла**

%macro task2 2

mov ah, 3Ch ;Функция DOS 3Ch (создание файла)

mov dx, %1 ;Указатель на начало строки в памяти (имя файла)

mov si, %2

mov cx,0 ;Нет атрибутов - обычный файл

call createfile;

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

task2 file\_name, message

;---[ Стандартное завершение программы ]----------------------—

mov AX, 4C00h

int 21h

; Процедура вывода сообщения об ошибке

error\_msg:

mov ah,9

mov dx,s\_error

int 21h ;Вывод сообщения об ошибке

ret

;==================================================================

createfile:

mov ah,3Ch ;Функция DOS 3Ch (создание файла)

mov dx,file\_name ;Имя файла

mov cx,0 ;Нет атрибутов - обычный файл

int 21h ;Обращение к функции DOS

jnc NO\_ERR1 ;Если нет ошибки, то продолжаем

call error\_msg ;Иначе вывод сообщения об ошибке

jmp exit1 ;Выход из программы

NO\_ERR1:

mov [handle],ax ;Сохранение дескриптора файла

write\_text:

mov bx,[handle] ;Дескриптор файла

mov ah,40h ;Функция DOS 40h (запись в файл)

mov dx,si ;Адрес строки с данными

movzx cx,[size] ;Размер данных

int 21h ;Обращение к функции DOS

jnc close\_file1 ;Если нет ошибки, то закрыть файл

call error\_msg ;Вывод сообщения об ошибке

close\_file1:

mov ah,3Eh ;Функция DOS 3Eh (закрытие файла)

mov bx,[handle] ;Дескриптор

int 21h ;Обращение к функции DOS

jnc exit1 ;Если нет ошибки, то выход из программы

call error\_msg ;Вывод сообщения об ошибке

exit1:

mov ah,9

mov dx,s\_pak

int 21h ;Вывод строки 'All is ok!'

RET

;==================================================================

;===[ Начало сегмента данных ]==========================================

file\_name db 'new\_file.txt',0

message db 'HeLLo',13,10,'message!'

size db 15

s\_error db 'Error!',13,10,'$'

s\_pak db 'All is ok!$'

endline db 13,10,'$'

handle dw 1 ;Дескриптор файла

**3. Вывод содержимого файла**

%macro task3 2

mov ah, 3Ch ;Функция DOS 3Ch (создание файла)

mov dx, %1 ;Указатель на начало строки в памяти (имя файла)

mov si, %2

mov cx,0 ;Нет атрибутов - обычный файл

call read\_file;

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

task3 file\_name, buffer

;---[ Стандартное завершение программы ]----------------------—

mov AX, 4C00h

int 21h

; Процедура вывода сообщения об ошибке

error\_msg:

mov ah,9

mov dx,s\_error

int 21h ;Вывод сообщения об ошибке

ret

;==================================================================

read\_file:

mov ah,3Dh ;Функция DOS 3Dh (открытие файла)

mov al,0 ;Режим открытия - только чтение

mov dx,file\_name ;Имя файла

mov cx,0 ;Нет атрибутов - обычный файл

int 21h ;Обращение к функции DOS

jnc NO\_ERR2 ;Если нет ошибки, то продолжаем

call error\_msg ;Иначе вывод сообщения об ошибке

jmp exit2 ;Выход из программы

NO\_ERR2:

mov [handle],ax ;Сохранение дескриптора файла

mov bx,[handle] ;Дескриптор файла

mov ah,3Fh ;Функция DOS 3Fh (чтение из файла)

mov dx,si ;Адрес буфера для данных

mov cx,80 ;Максимальное кол-во читаемых байтов

int 21h ;Обращение к функции DOS

jnc MYREAD2 ;Если нет ошибки, то продолжаем

call error\_msg ;Вывод сообщения об ошибке

jmp close\_file ;Закрыть файл и выйти из программы

MYREAD2:

mov bx, buffer

add bx,ax ;В AX количество прочитанных байтов

mov byte[bx],'$' ;Добавление символа '$'

mov ah,9

mov dx, buffer

int 21h ;Вывод содержимого файла

mov dx,endline

int 21h ;Вывод перехода на новую строку

close\_file:

mov ah,3Eh ;Функция DOS 3Eh (закрытие файла)

mov bx,[handle] ;Дескриптор

int 21h ;Обращение к функции DOS

jnc exit2 ;Если нет ошибки, то выход из программы

call error\_msg ;Вывод сообщения об ошибке

exit2:

mov ah,9 ;Вывод строки 'All is ok!'

mov dx,s\_pak

int 21h

RET

;==================================================================

;===[ Начало сегмента данных ]==========================================

file\_name db 'new\_file.txt',0

message db 'HeLLo',13,10,'message!'

size db 15

s\_error db 'Error!',13,10,'$'

s\_pak db 'All is ok!$'

endline db 13,10,'$'

handle dw 1 ;Дескриптор файла

buffer: times 100 db 0

**4.**

%macro task4 3

mov ah, 3Ch ;Функция DOS 3Ch (создание файла)

mov dx, %1 ;Указатель на начало строки в памяти (имя файла)

mov si, %2

mov di, %3

mov cx, 0 ;Нет атрибутов - обычный файл

call read\_file

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

task4 file\_name, buffer, 2

;---[ Стандартное завершение программы ]----------------------—

mov AX, 4C00h

int 21h

; Процедура вывода сообщения об ошибке

error\_msg:

mov ah,9

mov dx,s\_error

int 21h ;Вывод сообщения об ошибке

ret

;==================================================================

read\_file:

mov al,0 ;Режим открытия - только чтение

int 21h ;Обращение к функции DOS

jnc NO\_ERR2 ;Если нет ошибки, то продолжаем

call error\_msg ;Иначе вывод сообщения об ошибке

jmp exit2 ;Выход из программы

NO\_ERR2:

mov [handle],ax ;Сохранение дескриптора файла

mov bx,[handle] ;Дескриптор файла

mov ah,3Fh ;Функция DOS 3Fh (чтение из файла)

mov dx,si ;Адрес буфера для данных

mov cx,80 ;Максимальное кол-во читаемых байтов

int 21h ;Обращение к функции DOS

jnc READ1 ;Если нет ошибки, то продолжаем

call error\_msg ;Вывод сообщения об ошибке

jmp close\_file ;Закрыть файл и выйти из программы

READ1:

mov bx,buffer

add bx,ax ;В AX количество прочитанных байтов

mov byte[bx],'$' ;Добавление символа '$'

mov bx, buffer

mov bp, bx

mov cx, 0

next:

mov bp, si

mov dx, [bx]

inc bx

cmp dx,'$'

jne print

cmp dx,13

jne next

mov si,bx

inc cx

cmp cx,di

jne next

print:

mov ah,9

mov dx,bp

int 21h ;Вывод содержимого файла

mov dx,endline

int 21h ;Вывод перехода на новую строку

close\_file:

mov ah,3Eh ;Функция DOS 3Eh (закрытие файла)

mov bx,[handle] ;Дескриптор

int 21h ;Обращение к функции DOS

jnc exit2 ;Если нет ошибки, то выход из программы

call error\_msg ;Вывод сообщения об ошибке

exit2:

mov ah,9 ;Вывод строки 'All is ok!'

mov dx,s\_pak

int 21h

RET

;==================================================================

;===[ Начало сегмента данных ]==========================================

file\_name db 'new\_file.txt',0

message db 'HeLLo',13,10,'message!'

size db 15

s\_error db 'Error!',13,10,'$'

s\_pak db 'All is ok!$'

endline db 13,10,'$'

handle dw 1 ;Дескриптор файла

buffer: times 100 db 0

buffer2: times 100 db 0

**5.**

%macro task5 2

mov ah, 3Ch ;Функция DOS 3Ch (создание файла)

mov dx, %1 ;Указатель на начало строки в памяти (имя файла)

mov si, %2

mov cx, 0 ;Нет атрибутов - обычный файл

call read\_file

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

task5 file\_name, buffer

;---[ Стандартное завершение программы ]----------------------—

mov AX, 4C00h

int 21h

; Процедура вывода сообщения об ошибке

error\_msg:

mov ah,9

mov dx,s\_error

int 21h ;Вывод сообщения об ошибке

ret

;==================================================================

read\_file:

mov al,0 ;Режим открытия - только чтение

int 21h ;Обращение к функции DOS

jnc NO\_ERR2 ;Если нет ошибки, то продолжаем

call error\_msg ;Иначе вывод сообщения об ошибке

jmp exit2 ;Выход из программы

NO\_ERR2:

mov [handle],ax ;Сохранение дескриптора файла

mov bx,[handle] ;Дескриптор файла

mov ah,3Fh ;Функция DOS 3Fh (чтение из файла)

mov dx,si ;Адрес буфера для данных

mov cx,80 ;Максимальное кол-во читаемых байтов

int 21h ;Обращение к функции DOS

jnc READ1 ;Если нет ошибки, то продолжаем

call error\_msg ;Вывод сообщения об ошибке

jmp close\_file ;Закрыть файл и выйти из программы

READ1:

mov bx,buffer

add bx,ax ;В AX количество прочитанных байтов

mov byte[bx],'$' ;Добавление символа '$'

mov bx, buffer

mov bp, bx

mov cx, 0

next:

mov bp, si

mov dx, [bx]

inc bx

cmp dx,'='

jne print

cmp dx,13

jne next

mov si,bx

inc cx

cmp cx,di

jne next

print:

mov ah,9

mov dx,bp

int 21h ;Вывод содержимого файла

mov dx,endline

int 21h ;Вывод перехода на новую строку

close\_file:

mov ah,3Eh ;Функция DOS 3Eh (закрытие файла)

mov bx,[handle] ;Дескриптор

int 21h ;Обращение к функции DOS

jnc exit2 ;Если нет ошибки, то выход из программы

call error\_msg ;Вывод сообщения об ошибке

exit2:

mov ah,9 ;Вывод строки 'All is ok!'

mov dx,s\_pak

int 21h

RET

;==================================================================

;===[ Начало сегмента данных ]==========================================

file\_name db 'file.ini',0

message db 'HeLLo',13,10,'message!'

size db 15

s\_error db 'Error!',13,10,'$'

s\_pak db 'All is ok!$'

endline db 13,10,'$'

handle dw 1 ;Дескриптор файла

buffer: times 100 db 0

buffer2: times 100 db 0