

# Лекция 17. Системный файловый сервис

---

Системный файловый сервис прерывания 21h обеспечивает:

- создание файла и открытие существующего;
- запись байтов в файл или чтение байтов из него;
- установка указателя позиции байта в файле;
- закрытие файла и другие функции

При создании /открытии файла операционная система закрепляет за ним **двухбайтный «логический номер»**. При закрытии файла номер освобождается.

- Файловые функции возвращают в CF признак ошибки :  
CF = 0 - успешное выполнение;  
CF = 1 - ошибка и в AX - код ошибки  
!! Проверять CF после вызова файлового сервиса!!

## Создание файла (функция 3Ch )

---

Создает новый файл с нулевым размером (в байтах). Если файл существовал ранее, данные в нем теряются.

- Вход: **DS:DX** - адрес строки, содержащей путь к файлу. Строка должна заканчиваться числовым нулевым байтом (ASCIIZ –строка)

**CX** - атрибуты файла в младшем байте

Байт атрибутов содержит признаки свойств файла:

00h – стандартные атрибуты; 01h - только для чтения;  
02h - скрытый и т.д

- Выход: **AX** - логический номер файла

## Пример: создать файл по заданному пути

---

```
pathname  db  'd:\asm\file.asm', 0 ; путь к файлу
handle    dw  ? ; для логического номера файла

mov cx, 0 ; стандартные атрибуты файла
lea dx, pathname
mov ah, 3ch
int 21h
jc short error ; ошибка (CF=1)
mov ds:handle, ax ; сохраним логический номер

error: . . . .
```

## Открытие файла ( функция 3Dh )

---

Существующий файл «открывается» для доступа. Указатель позиции в файле устанавливается на первый байт.

- Вход: **DS:DX** - адрес в памяти ASCIIZ-строки с путем к файлу ;  
**AL** - код доступа к файлу: 0 – для чтения, 1- для записи, 2 – для чтения/записи
  
- Выход: **AX** - логический номер файла

## Чтение/изменение атрибутов файла (функция 43h )

---

Сервис считывает или изменяет атрибуты файла

- Вход: **DS:DX** - адрес в памяти ASCIIZ-строки с путем к файлу
- AL** - признак:
  - 0** – чтение атрибутов в регистр CL
  - 1** - придать файлу атрибуты, заданные в CL

# Чтение байтов из файла или с клавиатуры ( функция 3Fh )

---

Универсальный сервис: пригоден для чтения байтов с клавиатуры (логический номер устройства - 0) или из файла.

Чтение из файла начинается с текущей позиции. Затем, указатель позиции сместится на число прочитанных байт.

- Вход: **BX** - логический номер файла/устройства  
**CX** - количество считываемых байтов  
**DS:DX** – адрес памяти для размещения байтов
  
- Выход: **AX** - количество **фактически прочитанных** байтов  
**!!** Значение в AX будет меньше CX, если был достигнут конец файла

## Использование функции 3Fh для ввода с клавиатуры

Требования:

- ВХ = 0 - логический номер клавиатуры
- Ввод должен завершаться нажатием «Enter»
- Макс. количество считываемых байтов – 127, включая «Enter»

Сравнение функций 3Fh и 0Ah для ввода с клавиатуры	
3Fh	0Ah
- Читает байты из буфера клавиатуры. Если буфер пуст, будет ждать нажатия. Следовательно:	Ожидает нажатия с клавиатуры
!! Нажатия на клавиатуре и чтение байтов можно разнести во времени	Контролирует количество нажатий по «плану нажатий»

## Пример:

---

На клавиатуре были нажаты 5 символов и Enter. Прочитать все коды **из буфера клавиатуры** в память.

Buf db 7 dup (?) ; область памяти для записи кодов

. . . . .

; подготовка параметров для вызова сервиса

mov bx, 0 ; логический № клавиатуры

lea dx, Buf ; адрес области Buf в dx

mov cx, 7 ; количество считываемых байт в cx

mov ah, 3fh

int 21h

# Запись в файл или символьное устройство ( функция 40h )

---

Производит запись байтов из памяти в файл, начиная с текущей позиции в файле. После записи указатель позиции в файле перемещается на число записанных байтов.

- Вход:     **BX** - логический № файла/устройства  
              **CX** - количество записываемых байтов  
              **DS:DX** – адрес памяти
  
- Выход:   **AX** - количество **фактически записанных** байтов  
              **!!** Если при вызове задать **CX=0**, то файл будет «укорочен» до текущего значения указателя.

## Сместить позицию в файле (функция 42h )

---

- Вход: **BX** - логический № файла  
**AL** - код, задающий начало смещения:  
**0** – от начала, **1**- от текущей позиции, **2** - от конца файла  
**CX (ст) и DX(мл)** – 4-х байтная **знаковая** величина смещения в байтах
- Выход: **DX (ст) и AX (мл)** - 4-х байтное текущее значение указателя позиции от начала файла

**!!** Функцию можно использовать для **определения размера файла**, сделав вызов с параметрами: **AL =2** и **CX-DX=0**

## Закрывать файл ( функция 3Eh )

---

- Вход: ВХ - логический номер файла

Пример  
содержательного  
алгоритма

Задача: Символы 3-й  
строки файла вывести  
на экран

