



# Работа с файлами




При открытии файла ОС автоматически строит блок управления файлом. Блок строится в рабочих областях и его адрес не известен пользовательской программе. Блок и соответствующий файл идентифицируются двухбайтовым номером, который ОС возвращает программе после открытия файла. Этот номер называется *файловым дескриптором*.




Во всех дальнейших обращениях к системным функциям используется дескриптор файла. Т.е. при открытии файла пользовательская программа сообщает ОС его имя и получает обратно номер, который служит логическим именем файла при всех дальнейших операциях с файлом.





Необходимая для работы с файлом информация сохраняется в рабочих областях. Можно указать не только устройство и имя файла, но и полную файловую спецификацию, включающую и путь к файлу. Таким образом доступны все файлы. Для всех системных функций, обслуживающих метод файлового дескриптора, файловая спецификация задается ASCIIZ цепочкой символов с максимальной длиной 63 байта. ASCIIZ цепочка - это цепочка ASCII символов, заканчивающаяся значением 00h.



В рабочих областях DOS предусмотрено место для 8 управляющих блоков файлов, открытых по методу дескриптора. Это принятое по умолчанию количество блоков можно увеличить при конфигурировании системы командой FILES в CONFIG.SYS. Для каждого управляющего блока необходимо 48 байтов.





DOS использует первые пять дескрипторов для специальных файлов:

0 - стандартный вход (CON)

1 - стандартный выход (CON)

2 - стандартное устройство для сообщений (CON)

3 - стандартное вспомогательное устройство  
(AUX=COM1)

4 - стандартный принтер PRN=LPT1

Поэтому первый свободный для пользовательских программ файловый дескриптор имеет номер 5.



Все функции OS для работы с файлами, каталогами и дисками можно разделить на несколько групп:


1. Создание, открытие и закрытие файла
2. Запись и чтение данных
3. Изменение характеристик файла
4. Поиск файла
5. Операции над каталогами
6. Операции над дисками



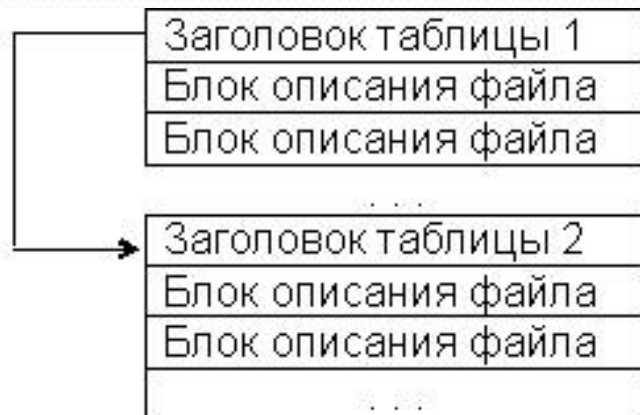
## ОТКРЫТИЕ И ЗАКРЫТИЕ ФАЙЛОВ

**Открыть файл** - это означает выделить для него фиксированную область памяти (блок управления), обнаружить файл и перенести определенную информацию из каталога в блок управления. Пользователь задает в качестве параметров функции OPEN имя и атрибуты файла, а DOS автоматически строит и заполняет блок управления для файла в своих рабочих областях. После выполнения OPEN пользователь получает двухбайтовый номер (дескриптор), которым файл идентифицируется во всех последующих операциях, вплоть до его закрытия. Неоткрытый файл дескриптора не имеет и система работать с ним не может.





Открывая файл, OS назначает ему очередной свободный элемент (блок описания файла) специальной системной таблицы, называемой таблицей открытых файлов (System File Table, SFT) и располагаемой в оперативной памяти среди системных областей данных.




Смещение	Число байтов	Описание
0	4	Адрес следующей таблицы или FFFFh в первом слове, если эта таблица последняя
4	2	Количество блокоов описания файлов в данной таблице

Найдя в системе каталогов диска запись об открываемом файле, DOS записывает в выделенный ему элемент SFT основные характеристики файла.



## БЛОК ОПИСАНИЯ ФАЙЛА


Смещение	Число байтов	Описание
00h	2	Количество дескрипторов, закрепленных за данным файлом или 0, если данный блок свободен
02h	2	Режим доступа к файлу
04h	1	Атрибуты файла
05h	2	Информационное слово устройства
07h	4	Указатель на заголовок драйвера символьного устройства или указатель на блок параметров дисководов
0Bh	2	Номер первого кластера файла
0Dh	2	Время последней модификации файла
0Fh	2	Дата последней модификации файла
11h	4	Размер файла в байтах
15h	4	Текущее положение указателя файла
19h	2	Относительный номер последнего прочитанного или записанного кластера файла
1Bh	4	Номер сектора с записью каталога о данном файле
1Fh	1	Номер записи каталога внутри сектора
20h	11	Имя и расширение файла
2Eh	2	Сегментный адрес PSP программы, открывшей файл
30h	2	Зарезервировано
32h	2	Абсолютный адрес последнего записанного или прочитанного кластера файла
34h	4	Зарезервировано



При работе с файлом OS может изменять некоторые из полей блока описания файла: размер файла, текущая позиция и др. Пользовательская программа также может изменять поля блока задавая параметры некоторых системных функций.

При всех дисковых операциях OS использует собственные рабочие области - буфера. Обмен с дисками осуществляется секторами (512 байт) - по одному сектору в каждом буфере. Системные функции ввода-вывода переносят записи файла из буферов OS в область пользовательской программы и обратно.





При открытии файл должен существовать.  
Указатель текущей позиции в файле - File  
pointer - устанавливается в начало файла. в AL  
задаются режимы.

7 6 5 4 3 2 1 0	НАЗНАЧЕНИЕ
. . . . . Д Д Д	Режим доступа
. . . . . Р . . . .	Резервный
. П П П . . . . .	Режим совместного использования
Н . . . . .	Режим наследования

## Режим доступа - биты 0-2.

Определяют права доступа программы к файлу:

(000) - только для чтения;

(001) - только для записи;

(010) - для чтения и записи.

Другие комбинации не допускаются.

7 6 5 4 3 2 1 0	НАЗНАЧЕНИЕ
. . . . . Д Д Д	Режим доступа
. . . . . Р . . . .	Резервный
. П П П . . . . .	Режим совместного использования
Н . . . . .	Режим наследования



## **Режим совместного использования - биты 4-6.**

Определяют право доступа к файлу для других процессов, которые пытаются открыть файл до его закрытия главным процессом, OS следит за соблюдением режима совместного использования файлов, только если в памяти загружен специальный резидентный модуль. Один и тот же файл можно открыть многократно в одном и том же процессе. При каждом открытии создается новый блок управления и получается новый файловый дескриптор.

7 6 5 4 3 2 1 0	НАЗНАЧЕНИЕ
. . . . . Д Д Д	Режим доступа
. . . . . Р . . . .	Резервный
. П П П . . . . .	Режим совместного использования
Н . . . . .	Режим наследования

## Режим наследования - бит 7

Указывает, как файл может быть использован порожденным процессом.

- ❖ бит 7 равен 0 - порожденный процесс наследует все открытые родительским процессом файлы в их текущем состоянии и с тем же режимом доступа.
- ❖ бит 7 равен 1 - порожденный процесс выполняется как независимая программа и может работать с файлом в соответствии с режимом совместного использования, заданным при его открытии родительским процессом.

7 6 5 4 3 2 1 0	НАЗНАЧЕНИЕ
. . . . . Д Д Д	Режим доступа
. . . . . Р . . . .	Резервный
. П П П . . . . .	Режим совместного использования
Н . . . . .	Режим наследования



## **Заккрытие файла**

**Заккрытие файла** - прекращение взаимодействия между блоком управления и файлом. При закрытии файла DOS очищает все выходные буферы, которые содержат записи этого файла, независимо от того заполнены они или нет. Таким образом обеспечивается целостность данных в файле. Файловый дескриптор и соответствующий блок управления освобождаются. При этом информация в каталоге заменяется информацией из блока управления.


## ЧТЕНИЕ И ЗАПИСЬ

Есть два основных вида доступа к данным файла:

1. Последовательный
2. Прямой


С точки зрения файловой организации каждый файл представляет собой непрерывную последовательность байтов. Нигде нет индикации, определяющей файл как последовательный или прямой. Нигде также не хранится информация о длине записи или способе их объединения в блоки. Однако, благодаря организации FAT, можно вычислить точное расположение любого байта файла - номер сектора и смещение в секторе - именно это дает возможность DOS осуществлять прямой доступ к данным файла.





Пользовательская программа выбирает вид доступа в соответствии с логикой обработки и логической структурой данных.


- Если файл состоит из записей разной длины, его необходимо обрабатывать последовательно.
- Если файл состоит из записей фиксированной длины, можно использовать как прямой , так и последовательный доступ к файлу или использовать комбинацию из двух методов доступа.

- 
- При обработке файлов с переменной длиной записи проблема заключается в ограничении отдельных записей, т.к. OS не поддерживает информации о длине логической записи.

Один из методов решения этого вопроса предусматривает вставку специальных символов-разделителей.


Другой метод заключается в наличии в каждой записи префикса, который содержит ее длину.



- 
- Для файлов с постоянной длиной записи достаточно знать очередной номер логической записи для вычисления ее местоположения в файле и осуществления прямого чтения или записи.

Организовать прямой доступ к файлу позволяет наличие указателя файла в SFT.

**Указатель файла** - это номер байта относительно начала файла, с которого начнется операция записи или чтения.



При работе с диском на уровне OS к файлам и каталогам можно обращаться только по именам. Местоположение файла на диске остается неизвестным. Нельзя обратиться к системным областям диска: загрузочным записям, FAT, каталогам. Для работы с этими областями, а также конкретными секторами и кластерами файлов используются либо функции драйвера BIOS (INT 13h), либо два специальных прерывания DOS - INT 25h и INT 26h, которые осуществляют доступ не к файлам, а к секторам диска.