

## Файл

Файл<sup>1</sup> — концепция в вычислительной технике: сущность, позволяющая получить доступ к какому-либо ресурсу вычислительной системы и обладающая рядом признаков:

- ❖ фиксированное имя (последовательность символов, число или что-то иное, однозначно характеризующее файл);
- ❖ определённое логическое представление и соответствующие ему операции чтения/записи.

Может быть любой — от последовательности бит (хотя читаем именно байтами, а точнее словами-группами из байт, по четыре, по восемь, по шестнадцать) до базы данных с произвольной организацией или любым промежуточным вариантом; многомерной базой данных, строго упорядоченной.

Первому случаю соответствуют операции чтения/записи потока и/или массива (то есть последовательные или с доступом по индексу), второму — команды СУБД. Промежуточные варианты — чтение и разбор всевозможных форматов файлов.

В информатике используют следующее определение: файл — поименованная последовательность байтов.

Работа с файлами реализуется средствами операционных систем.

Имена как у файлов имеют и обрабатываются похожим образом:

- ❖ области данных (необязательно на диске);
- ❖ устройства (как физические, порты например; так и виртуальные);
- ❖ потоки данных (в частности, вход или выход процесса) («fire» следует переводить словом «конвейер»);
- ❖ сетевые ресурсы, сокет;
- ❖ объекты операционной системы.

Файлы первого типа исторически возникли первыми и распространены наиболее широко, поэтому часто «файлом» называют и область данных, соответствующую имени.

## 1. Файловая система

Основная статья: файловая система

По мере развития вычислительной техники файлов в системах становилось всё больше. Для удобства работы с ними, их, как и другие данные, стали организовывать в структуры (тогда же появились символьные имена). Вначале это был простой массив, «привязанный» к конкретному носителю информации. В настоящее время наибольшее распространение получила древовидная организация с возможностью монтирования и вставки дополнительных связей (то есть ссылок). Соответственно, имя файла приобрело характер пути к файлу: перечисление узлов дерева файловой системы, которые нужно пройти, чтобы до него добраться.

## 2. Файл как объект API операционной системы

Операционная система предоставляет приложениям набор функций и структур для работы с файлами. Возможности операционной системы накладывают дополнительные ограничения на ограничения файловой системы. Подробности не являются предметом данной статьи, обратитесь к источникам, разработчикам операционных систем, специализированным сайтам. Например [[1]]

## 3. Свойства файла

В зависимости от файловой системы, файл может обладать различным набором свойств.

### Имя файла

В большинстве файловых систем имя файла используется для указания к какому именно файлу производится обращение. В различных файловых системах ограничения на имя файла сильно различаются:

---

<sup>1</sup>англ. file — скоросшиватель, напильник

- ❖ В FAT16 и FAT12 размер имени файла ограничен 8 символами (плюс 1 (точка) плюс 3 символа расширения).
- ❖ В VFAT ограничение 255 байт.
- ❖ В FAT32, HPFS имя файла ограничено 255 символами
- ❖ В NTFS имя ограничено в некоторых ОС 255 символами Unicode, по спецификации 32768 символов
- ❖ В ext2/ext3 ограничение 255 байт.

Помимо ограничений файловой системы, интерфейсы операционной системы дополнительно ограничивают набор символов, который допустим при работе с файлами.

Для MS-DOS в имени файла допустимы только заглавные латинские буквы, цифры. Недопустимы пробел, знак вопроса, звёздочка, символы больше/меньше, символ вертикальной черты.<sup>[1]</sup> При вызове системных функций именами файлов в нижнем или смешанном регистре, они приводятся к верхнему регистру.

Для Microsoft Windows в имени файла разрешены заглавные и строчные буквы, цифры, некоторые знаки препинания, пробел. Запрещены символы ><|?\*\/\ : ".

Для GNU/Linux (с учётом возможности маскировки) разрешены все символы, кроме / и байта, значение которого равно нулю, то есть 0x00.

Большинство операционных систем требуют уникальности имени файла в одном каталоге, хотя некоторые системы допускают файлы с одинаковыми именами (например, при работе с ленточными накопителями) <sup>[2]</sup>.

## Расширение имени файла

Расширение имени файла (часто расширение файла или расширение) как самостоятельный атрибут файла существует в файловых системах FAT16, FAT32, NTFS, используемых операционными системами MS-DOS, DR-DOS, PC DOS, MS Windows и используется для определения типа файла. Оно позволяет системе определить, каким приложением следует открывать данный файл. По умолчанию в операционной системе Windows расширение скрыто от пользователя.

В остальных файловых системах расширение — условность, часть имени, отделённая самой правой точкой в имени.

## Атрибуты

В некоторых файловых системах, таких как NTFS, предусмотрены атрибуты (обычно это бинарное значение «да»/«нет», кодируемое одним битом). Практически атрибуты не влияют на возможность доступа к файлам, для этого в некоторых файловых системах существуют права доступа<sup>[3]</sup>.

Название атрибута	перевод	значение	файловые системы	операционные системы
READ ONLY	только для чтения	в файл запрещено писать	FAT32, FAT12, FAT16, NTFS, HPFS, VFAT	DOS, OS/2, Windows
SYSTEM	системный	критический для работы операционной системы файл	FAT32, FAT12, FAT16, NTFS, HPFS, VFAT	DOS, OS/2, Windows
HIDDEN	скрытый	файл скрывается от показа, пока явно не сказано обратное	FAT32, FAT12, FAT16,	DOS, OS/2, Windows

			NTFS, HPFS, VFAT	
ARCHIVE	архивный (требующий архивации)	файл изменён после резервного копирования или не был скопирован программами резервного копирования	FAT32, FAT12, FAT16, NTFS, HPFS, VFAT	DOS, OS/2, Windows
SUID	Установка пользовательского ID	выполнение программы от имени владельца	ext2	Unix-like
SGID	Установка группового ID	выполнение программы от имени группы (для каталогов: любой файл созданный в каталоге с установленным SGID, получит заданную группу-владельца)	ext2	Unix-like
StickyBit	липкий бит	предписывает ядру не выгружать завершившуюся программу из памяти сразу, а лишь спустя некоторое время, чтобы избежать постоянной загрузки с диска наиболее часто используемых программ	ext2	Unix-like

## Время

Для файла могут быть определены следующие временные метки:

- ❖ Время создания
- ❖ Время модификации
- ❖ Время последнего доступа

## Владелец и группа файла

В некоторых файловых системах предусмотрено указание на владельца файла, и группу владельца.

## Права доступа

В некоторых файловых системах предусмотрена возможность для ограничения доступа пользователей к содержимому файла

В UNIX-подобных операционных системах для файлов обычно выделяют три типа прав:

- ❖ Право на запись
- ❖ Право на чтение
- ❖ Право на выполнение

Каждое право задаётся отдельно для владельца, для группы и для всех остальных. ACL позволяют расширить этот список.

В операционных системах Windows NT при работе с файловой системой NTFS права доступа задаются явно для пользователей или групп (или наследуются от вышестоящих объектов).

Права в себя включают:

- ❖ Право на чтение
- ❖ Право на запись
- ❖ Право на исполнение
- ❖ Право на удаление
- ❖ Право на смену атрибутов и владельца
- ❖ Право на создание, удаление подпапок (для папок)

❖ **Право на чтение прав доступа**

Каждое право может быть задано как разрешением, так и запретом, запрет имеет больший приоритет, чем разрешение.

## 4. Операции с файлом

Условно можно выделить два типа операций с файлом — связанные с его открытием, и выполняющиеся без его открытия. Операции первого типа обычно служат для чтения/записи информации или подготовки к записи/чтению. Операции второго типа выполняются с файлом как с «объектом» файловой системы, в котором файл является мельчайшей единицей структурирования.

### Операции, связанные с открытием файла

В зависимости от операционной системы те или иные операции могут отсутствовать.

Обычно выделяют дополнительные сущности, связанные с работой с файлом:

- ❖ **хэндлер файла, или дескриптор (описатель).** При открытии файла (в случае, если это возможно), операционная система возвращает число (или указатель на структуру), с помощью которого выполняются все остальные файловые операции. По их завершению файл закрывается, а хэндлер теряет смысл.
- ❖ **файловый указатель.** Число, являющееся смещением относительно нулевого байта в файле. Обычно по этому адресу осуществляется чтение/запись в случае, если вызов операции чтения/записи не предусматривает указание адреса. При выполнении операций чтения/записи файловый указатель смещается на число прочитанных (записанных) байт. Последовательный вызов операций чтения таким образом позволяет прочитать весь файл, не заботясь о его размере.
- ❖ **файловый буфер.** Операционная система (и/или библиотека языка программирования) осуществляет кэширование файловых операций в специальном буфере (участке памяти). При закрытии файла буфер сбрасывается.
- ❖ **режим доступа.** В зависимости от потребностей программы, файл может быть открыт на чтение и/или запись. Кроме того, некоторые операционные системы (и/или библиотеки) предусматривают режим работы с текстовыми файлами. Режим обычно указывается при открытии файла.
- ❖ **режим общего доступа.** В случае многозадачной операционной системы возможна ситуация, когда несколько программ одновременно хотят открыть файл на запись и/или чтение. Для регуляции этого существуют режимы общего доступа, указывающие на возможность осуществления совместного доступа к файлу (например, файл в который производится запись может быть открыт для чтения другими программами — это стандартный режим работы log-файлов).

Открытие файла (обычно в качестве параметров передается имя файла, режим доступа и режим совместного доступа, а в качестве значения выступает файловый хэндлер или дескриптор), кроме того, обычно имеется возможность в случае открытия на запись указать на то, должен ли размер файла изменяться на нулевой.

**Закрытие файла.** В качестве аргумента выступает значение, полученное при открытии файла. При закрытии все файловые буферы сбрасываются.

**Запись** — в файл помещаются данные.

**Чтение** — данные из файла помещаются в область памяти.

**Перемещение указателя** — указатель перемещается на указанное число байт вперёд/назад или перемещается по указанному смещению относительно начала/конца. Не все файлы позволяют выполнение этой операции (например, файл на ленточном накопителе может не «уметь» перематываться назад).

**Сброс буферов** — содержимое файловых буферов с незаписанной в файл информацией записывается. Используется обычно для указания на завершение записи

логического блока (для сохранения данных в файле на случай сбоя).

Получение текущего значения файлового указателя.

## Операции, не связанные с открытием файла

Операции, не требующие открытия файла оперируют с его «внешними» признаками — размером, именем, положением в дереве каталогов. При таких операциях невозможно получить доступ к содержимому файла, файл является минимальной единицей деления информации.

В зависимости от файловой системы, носителя информации, операционной системой часть операций может быть недоступна.

Список операций с файлами

- ❖ Открытие для изменения файла
- ❖ Удаление файла
- ❖ Переименование файла
- ❖ Копирование файла
- ❖ Перенос файла на другую файловую систему/носитель информации
- ❖ Создание симлинка или хардлинка
- ❖ Получение или изменение атрибутов файла

## 5. Типы файлов

В различных операционных и/или файловых системах могут быть реализованы различные типы файлов; кроме того, реализация различных типов может различаться.

«Обыкновенный файл» — файл, позволяющий операции чтения, записи, перемещения внутри файла

Каталог<sup>2</sup> или директория — файл, содержащий записи о входящих в него файлах. Каталоги могут содержать записи о других каталогах, образуя древовидную структуру.

Жёсткая ссылка<sup>3</sup> — в общем случае, одна и та же область информации может иметь несколько имён. Такие имена называют жёсткими ссылками (хардлинками). После создания хардлинкасказать где «настоящий» файл, а где хардлинк невозможно, так как имена равноправны. Сама область данных существует до тех пор пока существует хотя бы одно из имён. Хардлинки возможны только на одном физическом носителе.

Символьная ссылка (симлинк, софтлинк) — файл, содержащий в себе ссылку на другой файл или директорию. Может ссылаться на любой элемент файловой системы, в том числе, и расположенный на другом физическом носителе.

## 6. Особенности реализации

В операционной системе UNIX процессы (обычно находятся в каталоге /proc) и устройства (/dev) представляются в виде файлов особого рода, что позволяет использовать некоторые файловые операции для манипуляции этими объектами.

В некоторых файловых системах (например, в файловой системе OS VAX VMS) файлы имеют версию, что позволяет открывать более старые варианты данного файла. В файловой системе Mac OS (HFS) у файлов есть два «потока»: поток данных (где хранится содержимое файла) и поток ресурсов, хранящий информацию о программе, предназначенной для открывания данного файла и, возможно, некоторую информацию для этой программы. В NTFS файл может содержать, кроме основного, сколько угодно именованных потоков.

## 7. Проблема точного определения понятия файл

<sup>2</sup>англ. directory — алфавитный справочник

<sup>3</sup>англ. hardlink, часто используется калька «хардлинк»

Так как файл в разных операционных системах обладает различным набором атрибутов, свойств и методов доступа, универсального определения, которое бы учитывало все особенности, сформулировано не было.

Вопрос «Что такое файл?» неоднократно вызывал в эхо-конференции ru.os.cmp войны флейма, из-за чего попал в FAQ конференции с формулировкой [2]: "Q48: что такое файл? A48: ОПЯТЬ?"

## 8. Список литературы

1. Fernando J. Corbató et al. «An Experimental Time-Sharing System.» May 3, 1962.
2. Jerome H. Saltzer *CTSS Technical Notes*. Project MIT-LCS-TR016
3. Обзор файловых систем FAT, HPFS и NTFS

## Оглавление

1.	Файловая система.....	1
2.	Файл как объект API операционной системы.....	1
3.	Свойства файла .....	1
	Имя файла .....	1
	Расширение имени файла .....	2
	Атрибуты.....	2
	Время .....	3
	Владелец и группа файла.....	3
	Права доступа .....	3
4.	Операции с файлом.....	4
	Операции, связанные с открытием файла.....	4
	Операции, не связанные с открытием файла.....	5
5.	Типы файлов.....	5
6.	Особенности реализации.....	5
7.	Проблема точного определения понятия файл .....	5
8.	Список литературы .....	6