

# Файл

Материал из Википедии — свободной энциклопедии

**Файл** (англ. *file*) — именованная область данных на носителе информации, используемая как базовый объект взаимодействия с данными в операционных системах.

Работа с файлами реализуется как правило с применением файловых систем, обеспечивающих организацию работы с файлами и абстракцию над носителями информации. Обычно выделяют исполняемые файлы (программы) и собственно файлы данных (например, текстовые файлы или медиаконтейнеры). Формат файла — способ организации данных внутри файла, позволяющий записывать в него информацию в соответствии с её смыслом и интерпретировать записанное. Многие операционные системы приравнивают к файлам и обрабатывают сходным образом и другие ресурсы; наиболее широкое использование понятия файла характерно для Unix-подобных операционных систем, реализующих концепцию «всё есть файл» (англ. *everything is a file*). В них файлами считаются не только области данных на тome, но и, в частности, физические устройства (порты или принтеры), виртуальные устройства (/dev/null, /dev/random, /dev/urandom и другие), потоки данных (например, именованные каналы), сетевые ресурсы, сокеты.

| Содержание                                      |
|---|
| <u>История</u>                                  |
| <u>Файловая система</u>                         |
| <u>Файл как объект API операционной системы</u> |
| <u>Свойства файла</u>                           |
| <u>Имя файла</u>                                |
| <u>Расширение имени файла</u>                   |
| <u>Основные атрибуты</u>                        |
| <u>Время</u>                                    |
| <u>Владелец и группа файла</u>                  |
| <u>Права доступа</u>                            |
| <u>Операции с файлом</u>                        |
| <u>Операции, связанные с открытием файла</u>    |
| <u>Операции, не связанные с открытием файла</u> |
| <u>Типы файлов</u>                              |
| <u>Особенности реализации</u>                   |
| <u>Примечания</u>                               |

## История

Слово *file* впервые применено к компьютерной системе хранения в 1950 году. Реклама памяти на запоминающих ЭЛТ фирмы RCA в журнале Popular Science<sup>[1]</sup> гласила:

«...результаты бесчисленных вычислений можно держать «в картотеке» (англ. *on file*) и получать снова. Эта «картотека» теперь существует в запоминающей трубке, разработанной в лабораториях RCA. Она электрически сохраняет цифры, отправленные в вычислительную машину, и держит их в хранилище, заодно запоминая новые — ускоряя интеллектуальные решения в лабиринтах математики.»



Файл перфокарт

В 1952 году слово *file* отнесли к колоде перфокарт.<sup>[2]</sup> Поначалу словом *file* называли само устройство памяти, а не его содержимое (регистровый файл). Например, диски IBM 350, использовавшиеся, например, в машине IBM 305, назывались *disk files*<sup>[3]</sup>. Системы наподобие Compatible Time-Sharing System ввели концепцию файловой системы, когда на одном запоминающем устройстве существует несколько виртуальных «устройств памяти», что и дало слову «файл» современное значение. Имена файлов в CTTS состояли из двух частей, «основного имени» и «дополнительного имени» (последнее существует и поныне как расширение имени файла)<sup>[4][5]</sup>.



Два жёстких диска машины IBM 305

## 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.

## 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. С точки зрения API файл — объект, по отношению к которому могут быть применены функции этого API. На уровне API уже не существенно, существует ли файл как объект файловой системы или является, например, устройством ввода-вывода.

## 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: в FAT16 и FAT12 размер имени файла ограничен 8.3 знаками (8 на имя и 3 на расширение); в других системах имя файла ограничено обычно в 255 байт; в NTFS имя ограничено в некоторых ОС 256 символами Unicode (по спецификации — 32 768 символов).

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 и меньше, символ вертикальной черты.<sup>[6]</sup> При вызове системных функций имена файлов в нижнем или смешанном регистре приводятся к верхнему регистру.

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

Для Linux (с учётом возможности маскировки) разрешены все символы, кроме / и *нуль-символа*.

Большинство операционных систем требуют уникальности имени файла в одном каталоге, хотя некоторые системы допускают файлы с одинаковыми именами (например, при работе с ленточными накопителями).

В некоторых файловых системах можно создавать и использовать файлы без имени. Такие файлы всегда являются временными. Их основные преимущества: при создании не может возникнуть конфликта имён и их нельзя найти в их каталоге по имени. В остальном это полноценные файлы.

## 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 операционных системах Windows и Mac OS расширение скрыто от пользователя.

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

## Основные атрибуты

В некоторых файловых системах, таких как NTFS, предусмотрены атрибуты (обычно это бинарное значение «да»/«нет», кодируемое одним битом). Во многих современных операционных системах атрибуты практически не влияют на возможность доступа к файлам, для этого в некоторых операционных и файловых системах существуют права доступа.

| Название атрибута | Перевод           | Значение                     | Файловые системы                    | Операционные системы  |
|-------------------|-------------------|------------------------------|-------------------------------------|-----------------------|
| Read only         | только для чтения | файл запрещено редактировать | FAT32,<br>FAT12,<br>FAT16,<br>NTFS, | DOS, OS/2,<br>Windows |

|                   |                                   |   |  |                       |
|-------------------|-----------------------------------|---|--|-----------------------|
|                   |                                   |   | HPFS,<br>VFAT  |                       |
| System            | системный                         | критический для работы операционной системы файл  | FAT32,<br>FAT12,<br>FAT16,<br>NTFS,<br>HPFS,<br>VFAT | DOS, OS/2,<br>Windows |
| Hidden            | скрытый                           | файл скрывается от показа, пока явно не указано обратное  | FAT32,<br>FAT12,<br>FAT16,<br>NTFS,<br>HPFS,<br>VFAT | DOS, OS/2,<br>Windows |
| Archive           | архивный<br>(требующий архивации) | файл изменён после резервного копирования или не был скопирован программами резервного копирования; при изменении файла ОС автоматически устанавливает этот атрибут   | FAT32,<br>FAT12,<br>FAT16,<br>NTFS,<br>HPFS,<br>VFAT | DOS, OS/2,<br>Windows |
| <u>SUID</u>       | Установка пользовательского ID    | выполнение программы от имени владельца   | ext2   | Unix-like             |
| <u>SGID</u>       | 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             |
| <u>Sticky Bit</u> | липкий бит                        | изначально предписывал ядру не выгружать завершившуюся программу из памяти сразу, а лишь спустя некоторое время, чтобы избежать постоянной загрузки с диска наиболее часто используемых программ, в настоящее время в разных ОС используется по-разному | ext2   | Unix-like             |

## Время

Для файла могут быть определены временные метки создания, последней модификации, последнего доступа и другие.

## 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 ранее обычно выделяли три типа прав: на запись, чтение и выполнение. Каждое право задаётся отдельно для владельца, для группы и для всех остальных. ACL позволяет использовать более детальное разделение и прав, и пользователей.

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, запись, исполнение, удаление, смену атрибутов и владельца, создание и удаление подпапок (для папок) и чтение прав доступа и другие как в POSIX. Каждое право может быть задано как разрешением, так и запретом. Запрет имеет больший приоритет, чем разрешение в отличие от POSIX.

## 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 и записи информации или подготовки к чтению или записи. Операции второго типа выполняются с файлом как с «объектом» файловой системы, в котором файл является наименьшим элементом 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:

- Дескриптор файла. 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 завершении файл закрывается, а дескриптор теряет смысл.
- Файловый указатель. Число, являющееся смещением относительно нулевого байта в файле. Обычно по этому адресу осуществляется чтение и запись, в случае, если вызов операции чтения или записи не предусматривает указание адреса. При выполнении операций чтения или записи файловый указатель увеличивается на число прочитанных или записанных байт. Последовательный вызов операций чтения таким образом позволяет прочитать весь файл последовательно, не заботясь о позиционировании.
- Файловый буфер. Операционная система 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 или запись. Кроме того, некоторые операционные системы и 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 чтение или запись. Для регулирования 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 с файлами регистрации).

## Операции

- Открытие файла (обычно в качестве параметров передается имя файла, режим доступа и режим совместного доступа, а в качестве результата выступает файловый 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 вперед или назад или перемещается по указанному смещению относительно начала или конца. Не все файлы позволяют выполнение этой операции (например, файл на ленточном накопителе может не «уметь» перематываться назад).
- Сброс буферов — содержимое файловых буферов с не записанной 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.

Возможные операции с файлами: создание, удаление, переименование, копирование, перенос на другую файловую систему, создание символьной ссылки или жёсткой ссылки, получение и изменение атрибутов.

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

## Типы файлов

---

По способу организации файлы делятся на файлы с произвольным доступом и файлы с последовательным доступом.

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

- «Обыкновенный файл» — файл, позволяющий операции чтения, записи, позиционирования внутри файла, изменения размера, иногда работу с атрибутами.
  - Временный файл — усиленное кэширование с запоздалой записью на диск; иногда — удаление сразу после того, как файл закрывается.

- **Каталог** (англ. *directory* — алфавитный справочник) или **директория** (также «папка») — файл, содержащий записи о входящих в него файлах. Каталоги могут содержать записи о других каталогах, образуя древовидную структуру, а при наличии ссылок — сетевую структуру.
- **Жёсткая ссылка** (англ. *hardlink*) — в общем случае, одна и та же область информации может иметь несколько имён. Такие имена называют жёсткими ссылками (хардлинками). После создания жёсткой ссылки сказать, где «настоящий» файл, а где жёсткая ссылка, невозможно, так как имена равноправны. Сама область данных существует до тех пор, пока существует хотя бы одно из имён. Жёсткие ссылки возможны только в пределах одного экземпляра файловой системы.
- **Символьная ссылка** (символическая ссылка) — файл, содержащий в себе ссылку на имя нужного файла любого типа. Может ссылаться на любой элемент файловой системы, в том числе, и расположенный на другом физическом носителе. В некоторых ФС может ссылаться на несуществующий файл, образовывать циклы, ссылаться на себя.
- **Приравниваемые к файлам объекты** — например, устройства.

## Особенности реализации



---

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

В некоторых файловых системах (например, в файловой системе 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.

## Примечания

---

1. *Bonnier Corporation. Popular Science* (<https://books.google.com/books?id=9SwDAAAAMBAJ&pg=PA96>). — Bonnier Corporation, 1950. — С. 96—.
2. Robert S. Casey, et al. *Punched Cards: Their Applications to Science and Industry*, 1952.
3. Martin H. Weik. Ballistic Research Laboratories Report #1115. March 1961. pp. 314—331 (<http://ed-thelen.org/comp-hist/BRL61-ibm03.html#IBM-305-RAMAC>)
4. Fernando J. Corbató et al. «An Experimental Time-Sharing System (<http://larch-www.lcs.mit.edu:8001/~corbato/sjcc62/>).» May 3, 1962.
5. Jerome H. Saltzer *CTSS Technical Notes* (<http://www.lcs.mit.edu/publications/pubs/pdf/MIT-LCS-TR-016.pdf>)  *Архивная копия* (<http://web.archive.org/web/20060913210311/http://www.lcs.mit.edu/publications/pubs/pdf/MIT-LCS-TR-016.pdf>)  *от 13 сентября 2006 на Wayback Machine*. Project MIT-LCS-TR016
6. Обзор файловых систем FAT, HPFS и NTFS (<http://support.microsoft.com/kb/100108/ru/>)

---

Источник — <https://ru.wikipedia.org/w/index.php?title=Файл&oldid=113925880>

---

Эта страница в последний раз была отредактирована 30 апреля 2021 в 18:30.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.

