

## **Файловые системы**

Файловая система (ФС) — это часть операционной системы, включающая:

- совокупность всех файлов на диске;
- наборы структур данных, используемых для управления файлами, такие, например, как каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске;
- комплекс системных программных средств, реализующих различные операции над файлами, такие как создание, уничтожение, чтение, запись, именование и поиск файлов.

Файловая система позволяет программам обходиться набором достаточно простых операций для выполнения действий над некоторым абстрактным объектом, представляющим файл. При этом программистам не нужно иметь дело с деталями действительного расположения данных на диске, буферизацией данных и другими низкоуровневыми проблемами передачи данных с долговременного запоминающего устройства. Все эти функции файловая система берет на себя. Файловая система распределяет дисковую память, поддерживает именование файлов, отображает имена файлов в соответствующие адреса во внешней памяти, обеспечивает доступ к данным, поддерживает разделение, защиту и восстановление файлов.

Таким образом, файловая система играет роль промежуточного слоя, экранирующего все сложности физической организации долговременного хранилища данных, и создающего для программ более простую логическую модель этого хранилища, а также предоставляя им набор удобных в использовании команд для манипулирования файлами.

**Задачи, решаемые ФС, зависят от способа организации вычислительного процесса в целом.** Самый простой тип - это ФС в однопользовательских и однопрограммных ОС, к числу которых относится, например, MS-DOS. Основные функции в такой ФС нацелены на решение следующих задач:

- именование файлов;
- программный интерфейс для приложений;
- отображения логической модели файловой системы на физическую организацию хранилища данных;
- устойчивость файловой системы к сбоям питания, ошибкам аппаратных и программных средств.

Задачи ФС усложняются в операционных однопользовательских мультипрограммных ОС, которые, хотя и предназначены для работы одного пользователя, но дают ему возможность запускать одновременно несколько процессов. Одной из первых ОС этого типа стала OS/2. К перечисленным выше задачам добавляется новая задача совместного доступа к файлу из нескольких процессов. Файл в этом случае является разделяемым ресурсом, а значит, файловая система должна решать весь комплекс проблем, связанных с такими ресурсами. В частности, в ФС должны быть предусмотрены средства блокировки файла и его частей, предотвращения гонок, исключение тупиков, согласование копий и т. п.

В многопользовательских системах появляется еще одна задача: защита файлов одного пользователя от несанкционированного доступа другого пользователя.

Еще более сложными становятся функции ФС, которая работает в составе сетевой ОС.

## Типы файлов

Файловые системы поддерживают несколько функционально различных типов файлов, в число которых, как правило, входят обычные файлы, файлы-каталоги, специальные файлы, именованные конвейеры, отображаемые в память файлы и другие.

**Обычные файлы**, или просто файлы, содержат информацию произвольного характера, которую заносит в них пользователь или которая образуется в результате работы системных и пользовательских программ. Большинство современных операционных систем (например, UNIX, Windows, OS/2) никак не ограничивает и не контролирует содержимое и структуру обычного файла. Содержание обычного файла определяется приложением, которое с ним работает. Например, текстовый редактор создает текстовые файлы, состоящие из строк символов, представленных в каком-либо коде. Это могут быть документы, исходные тексты программ и т. п. Текстовые файлы можно прочитать на экране и распечатать на принтере. Двоичные файлы не используют коды символов, они часто имеют сложную внутреннюю структуру, например исполняемый код программы или архивный файл. Все операционные системы должны уметь распознавать хотя бы один тип файлов — их собственные исполняемые файлы.

**Каталоги** — это особый тип файлов, которые содержат системную справочную информацию о наборе файлов, сгруппированных пользователями по какому-либо неформальному признаку (например, в одну группу объединяются файлы, содержащие документы одного договора, или файлы, составляющие один программный пакет). Во многих операционных системах в каталог могут входить файлы любых типов, в том числе другие каталоги, за счет чего образуется древовидная структура, удобная для поиска. Каталоги устанавливают соответствие между именами файлов и их характеристиками, используемыми файловой системой для управления файлами. В число таких

характеристик входит, в частности, информация (или указатель на другую структуру, содержащую эти данные) о типе файла и расположении его на диске, правах доступа к файлу и датах его создания и модификации. Во всех остальных отношениях каталоги рассматриваются файловой системой как обычные файлы.

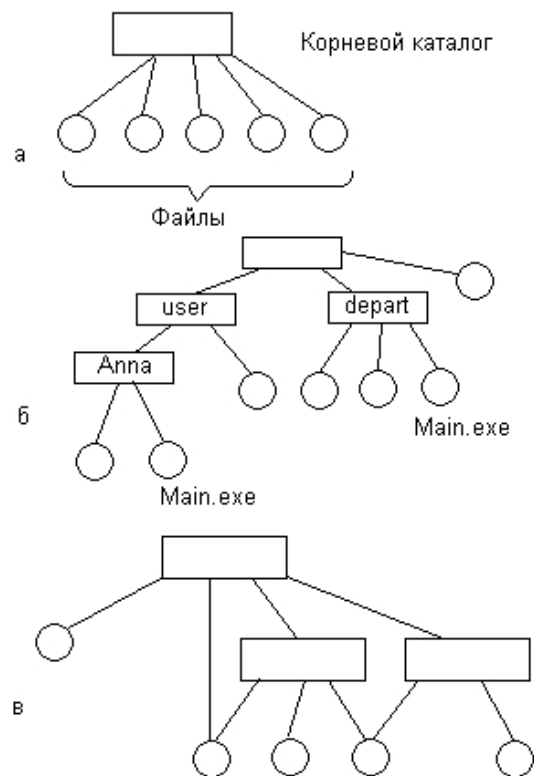
**Специальные файлы** — это фиктивные файлы, ассоциированные с устройствами ввода-вывода, которые используются для унификации механизма доступа к файлам и внешним устройствам. Специальные файлы позволяют пользователю выполнять операции ввода-вывода посредством обычных команд записи в файл или чтения из файла. Эти команды обрабатываются сначала программами файловой системы, а затем на некотором этапе выполнения запроса преобразуются операционной системой в команды управления соответствующим устройством.

Современные файловые системы поддерживают и другие типы файлов, такие как:

- **символьные связи,**
- **именованные конвейеры,**
- **отображаемые в память файлы.**

## Иерархическая структура файловой системы

Пользователи обращаются к файлам по символьным именам. Однако способности человеческой памяти ограничивают количество имен объектов, к которым пользователь может обращаться по имени. Иерархическая организация пространства имен позволяет значительно расширить эти границы. Именно поэтому большинство файловых систем имеет иерархическую структуру, в которой уровни создаются за счет того, что каталог более низкого уровня может входить в каталог более высокого уровня (рис. 1).



**Рис. 1.** Иерархия файловых систем

Граф, описывающий иерархию каталогов, может быть деревом или сетью. Каталоги образуют дерево, если файлу разрешено входить только в один каталог (рис. 1, б), и сеть — если файл может входить сразу в несколько каталогов (рис. 1, в). Например, в MS-DOS и Windows каталоги образуют древовидную структуру, а в UNIX — сетевую. В древовидной структуре каждый файл является листом. Каталог самого верхнего уровня называется корневым каталогом, или корнем (root).

При такой организации пользователь освобожден от запоминания имен всех файлов, ему достаточно примерно представлять, к какой группе может быть отнесен тот или иной файл, чтобы путем последовательного просмотра каталогов найти его. Иерархическая структура удобна для многопользовательской работы: каждый пользователь со своими файлами локализуется в своем каталоге или поддереве каталогов, и вместе с тем все файлы в системе логически связаны.

Частным случаем иерархической структуры является одноуровневая организация, когда все файлы входят в один каталог (рис. 1, а).

Все типы файлов имеют символьные имена. В иерархически организованных файловых системах обычно используются три типа имен файлов:

- **простые,**
- **составные,**
- **относительные.**

Простое, или короткое, символьное имя идентифицирует файл в пределах одного каталога. Простые имена присваивают файлам пользователи и программисты, при этом они должны учитывать ограничения ОС как на номенклатуру символов, так и на длину имени. До сравнительно недавнего времени эти границы были весьма узкими. Так, в популярной файловой системе FAT длина имен ограничивались схемой 8.3 (8 символов — собственно имя, 3 символа — расширение имени), а в файловой системе s5, поддерживаемой многими версиями ОС UNIX, простое символьное имя не могло содержать более 14 символов.

Однако пользователю гораздо удобнее работать с длинными именами, поскольку они позволяют дать файлам легко запоминающиеся названия, ясно говорящие о том, что содержится в этом файле. Поэтому современные файловые системы, а также усовершенствованные варианты уже существовавших файловых систем, как правило, поддерживают длинные простые символьные имена файлов. Например, в файловых системах NTFS и FAT32, входящих в состав операционной системы Windows NT, имя файла может содержать до 255 символов.

Примеры простых имен файлов и каталогов:

Имя файла.doc  
task-entran.exe

В иерархических файловых системах разным файлам разрешено иметь одинаковые простые символьные имена при условии, что они принадлежат разным каталогам. То есть здесь работает схема **«много файлов — одно простое имя»**. Для однозначной идентификации файла в таких системах используется так называемое полное имя.

Полное имя представляет собой цепочку простых символьных имен всех каталогов, через которые проходит путь от корня до данного файла. Таким образом, полное имя является составным, в котором простые имена отделены друг от друга принятым в ОС разделителем. Часто в качестве разделителя используется прямой или обратный слеш, при этом принято не указывать имя корневого каталога. На рис. 7.3, б два файла имеют простое имя main.exe, однако их составные имена /depart/main.exe и /user/anna/main.exe различаются.

В древовидной файловой системе между файлом и его полным именем имеется взаимно однозначное соответствие «один файл — одно полное имя». В файловых системах, имеющих сетевую структуру, файл может входить в несколько каталогов, а значит, иметь несколько полных имен; здесь справедливо соответствие «один файл — много полных имен». В обоих случаях файл однозначно идентифицируется полным именем.

Файл может быть идентифицирован также относительным именем. Относительное имя файла определяется через понятие «текущий каталог». Для каждого пользователя в каждый момент времени один из каталогов файловой системы является текущим, причем этот каталог выбирается самим пользователем по команде ОС. Файловая система фиксирует имя текущего каталога, чтобы затем использовать его как дополнение к относительным именам для образования полного имени файла. При использовании относительных имен пользователь идентифицирует файл цепочкой имен каталогов, через которые проходит маршрут от текущего каталога до данного файла. Например, если

текущим каталогом является каталог /user, то относительное имя файла /user/anna/main.exe выглядит следующим образом: anna/ main.exe.

В некоторых операционных системах разрешено присваивать одному и тому же файлу несколько простых имен, которые можно интерпретировать как псевдонимы. В этом случае, так же как в системе с сетевой структурой, устанавливается соответствие «один файл — много полных имен», так как каждому простому имени файла соответствует по крайней мере одно полное имя.

И хотя полное имя однозначно определяет файл, операционной системе проще работать с файлом, если между файлами и их именами имеется взаимно однозначное соответствие. С этой целью она присваивает файлу уникальное имя, так что справедливо соотношение «один файл — одно уникальное имя». Уникальное имя существует наряду с одним или несколькими символьными именами, присваиваемыми файлу пользователями или приложениями. Уникальное имя представляет собой числовой идентификатор и предназначено только для операционной системы. Примером такого уникального имени файла является номер индексного дескриптора в системе UNIX.

## Монтирование

В общем случае вычислительная система может иметь несколько дисковых устройств. Даже типичный персональный компьютер обычно имеет один накопитель на жестком диске, один накопитель на гибких дисках и накопитель для компакт-дисков. Мощные же компьютеры, как правило, оснащены большим количеством дисковых накопителей, на которые устанавливаются пакеты дисков. Более того, даже одно физическое устройство с помощью средств операционной системы может быть представлено в виде нескольких логических устройств, в частности путем разбиения дискового пространства на разделы. Возникает вопрос, каким образом организовать хранение файлов в системе, имеющей несколько устройств внешней памяти?

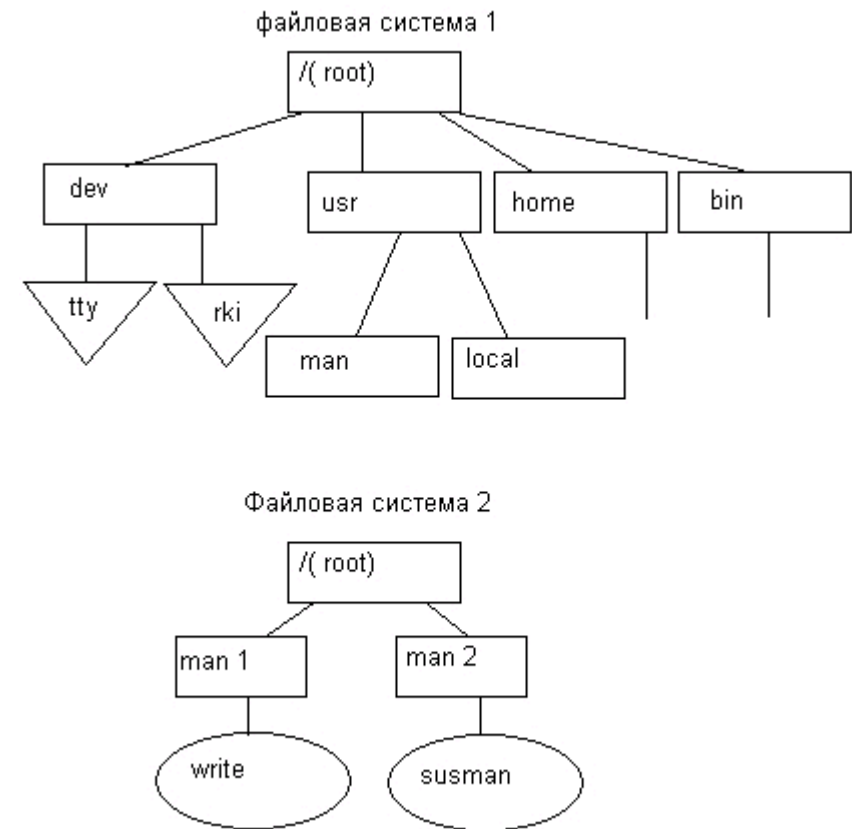
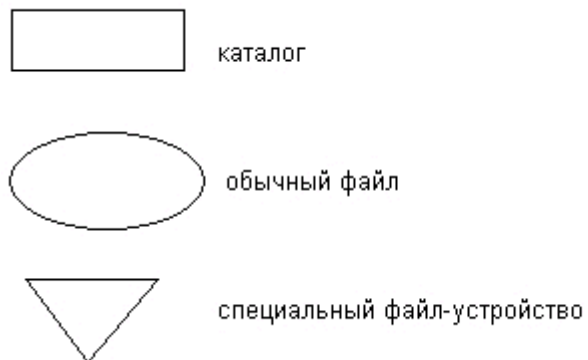
Первое решение состоит в том, что на каждом из устройств размещается автономная файловая система, то есть файлы, находящиеся на этом устройстве, описываются деревом каталогов, никак не связанным с деревьями каталогов на других устройствах. В таком случае для однозначной идентификации файла пользователь наряду с составным символьным именем файла должен указывать идентификатор логического устройства. Например, [D:\docs\privat\letter\uni\let1.doc](#)

**Монтирование** – это логическое объединение файловых систем, расположенных на разных физических устройствах.

При монтировании файловая система, расположенная на системном диске, назначается корневой. Для связи иерархий файлов в корневой файловой системе выбирается некоторый существующий каталог, в данном примере — каталог `man`. (рис. 1). После выполнения монтирования выбранный каталог `man` становится корневым каталогом второй файловой системы. Через этот каталог монтируемая файловая система подсоединяется как поддерево к общему дереву (рис. 2).

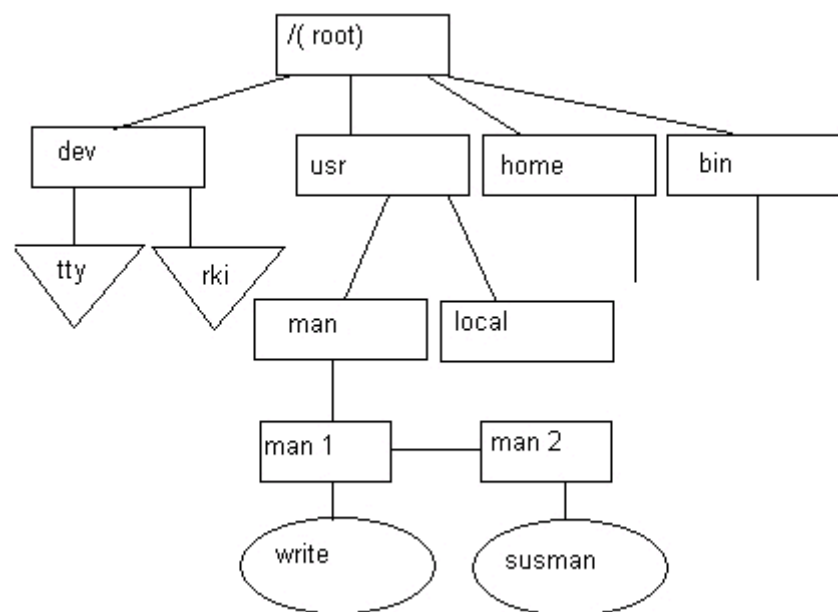
После монтирования общей файловой системы для пользователя нет логической разницы между корневой и смонтированной файловыми системами, в частности именование файлов производится так же, как если бы она с самого начала была единой.

#### Обозначения:



**Рис. 1.** Две файловые системы до монтирования

**Рис. 2.** Общая файловая система после монтирования



Дата: " \_\_\_\_ " \_\_\_\_\_ 2012 г.

**FS. пр [2]**

ФИО: \_\_\_\_\_ Группа ФИТ- \_\_\_\_\_

№	Вопрос	Ответ
1	Что понимается под файловой системой (ФС) с точки зрения диска?	
2	Верно ли утверждение, что средствами дополнительного ПО на одном HDD не возможно создать пять и более разделов?	
3	Что такое монтирование?	
4	Что такое каталог?	
5	Верно ли утверждение, что без ФС можно обойтись, если использовать специализированное ПО?	
6	В каком случае граф, описывающий иерархию каталогов, может быть назван деревом?	
7	Как называется каталог самого верхнего уровня?	
8	Верно ли утверждение, что большинство современных ОС никак не ограничивает и не контролирует содержимое и структуру обычного файла?	
9	Перечислите достоинства иерархической структуры ФС	
10	Кратко охарактеризуйте относительный тип имен файлов	
11	Верно ли утверждение, что ФС не может играть роль промежуточного слоя, экранирующего все сложности физической организации?	
12	От чего могут зависть задачи, решаемые ФС?	
13	Укажите особенности составного типа имен файлов	
14	Верно ли утверждение, что возможно создание одного логического диска из освободившегося пространства после удаления D:\ и E:\, <b>ОБОСНУЙТЕ</b>	
15	Каким образом можно разместить на одном логическом диске две различные файловые системы?	