**Проблема:**

* Приложениям требуется хранить больше данных, чем помещается в виртуальном адресном пространстве.
* Информация, хранящаяся в адресном пространстве, теряется при завершении процесса.
* Необходим одновременный доступ к информации из разных процессов.

**Решение:**

* Файловая система - это абстракция, которая позволяет хранить информацию на долговременных носителях (магнитных дисках, SSD).
* Файлы - это логические информационные блоки, создаваемые процессами.
* Файловая система обеспечивает:
  + Долговременное хранение информации.
  + Независимость от процессов.
  + Одновременный доступ из разных процессов.

**Структура:**

* **Пользовательский интерфейс:**
  + Работа с файлами.
  + Имена файлов.
  + Защита.
  + Доступные операции.
* **Реализация:**
  + Отслеживание свободного пространства.
  + Организация данных на диске.
  + Управление файловой системой.
* **Примеры файловых систем:**
  + FAT.
  + NTFS.
  + EXT4.

**Важно:**

* Файловая система - это ключевая часть операционной системы.
* Понимание работы файловой системы важно для специалистов по ОС.
* Пользователю важно знать интерфейс работы с файлами.

**1.1. Имена файлов:**

* Имя файла - это механизм абстрагирования, который позволяет сохранить информацию на диске и затем ее считать.
* Правила составления имен файлов varьируются, но обычно допускаются от 1 до 8 букв, цифры и специальные символы.
* Некоторые файловые системы различают буквы верхнего и нижнего регистров, другие нет.
* Расширение имени файла - это часть имени, которая следует за точкой и обычно несет в себе информацию о типе файла.
* Расширения могут быть обязательными или необязательными, и их использование может быть соглашением или навязываться операционной системой.

**1.2. Структура файла:**

* Файлы могут быть организованы как последовательность байтов, записей или дерева записей.
* Последовательность байтов - это наиболее простая структура, но она не обеспечивает никакой структуры.
* Последовательность записей фиксированной длины обеспечивает более удобную структуру, позволяя считывать и записывать отдельные записи.
* Дерево записей - это наиболее сложная структура, но она позволяет выполнять быстрый поиск по ключу.

**1.3. Типы файлов:**

* Операционные системы обычно поддерживают несколько типов файлов, например, обычные файлы, каталоги, специальные файлы.
* Обычные файлы содержат пользовательскую информацию.
* Каталоги - это системные файлы, которые используются для поддержки структуры файловой системы.
* Специальные файлы используются для моделирования устройств ввода-вывода, таких как терминалы, принтеры и сети.

**1.4. Доступ к файлам:**

* Существует два основных типа доступа к файлам: последовательный и произвольный.
* Последовательный доступ позволяет читать или записывать байты или записи файла только в порядке их следования.
* Произвольный доступ позволяет читать или записывать байты или записи файла в любом порядке.

**1.5. Атрибуты файлов:**

* Каждый файл имеет атрибуты, которые содержат дополнительную информацию о файле, например, его размер, дату создания, права доступа.
* Атрибуты могут быть использованы для защиты файлов, управления доступом к ним, отслеживания изменений.

**1.6. Операции с файлами:**

* Операционные системы предоставляют различные операции для работы с файлами, например, создание, удаление, открытие, закрытие, чтение, запись, поиск.
* Эти операции позволяют создавать, модифицировать, удалять и просматривать информацию, хранящуюся в файлах.

**Дополнительно:**

* Текст также содержит информацию о различных файловых системах, таких как FAT, NTFS, exFAT.
* В тексте также приводятся примеры структуры двоичных файлов, таких как исполняемые файлы и архивы.

Файловая система - это важный компонент операционной системы, который позволяет ей управлять хранением информации на диске. Файловая система предоставляет различные механизмы для организации, защиты и доступа к информации, хранящейся на диске.

**Каталоги**

**1. Иерархическая система каталогов:**

* Каталоги (папки) – это файлы, которые используются для упорядочивания других файлов.
* Иерархическая система каталогов позволяет группировать файлы по тематике.
* Файлы могут иметь абсолютные или относительные имена.
* Абсолютные имена начинаются с корневого каталога.
* Относительные имена определяются относительно текущего рабочего каталога.
* В каждом каталоге есть специальные элементы "." (текущий каталог) и ".." (родительский каталог).

**2. Операции с каталогами:**

* **Create:** создание пустого каталога.
* **Delete:** удаление пустого каталога.
* **Opendir:** открытие каталога для чтения.
* **Closedir:** закрытие каталога после чтения.
* **Readdir:** чтение следующей записи из открытого каталога.
* **Rename:** переименование каталога.
* **Link:** создание жесткой ссылки (привязки) к файлу, позволяющей ему фигурировать в нескольких каталогах.
* **Unlink:** удаление записи каталога, что может привести к удалению файла, если он не используется нигде другом.
* **Symbolic link:** создание символической ссылки, которая представляет собой небольшой файл, содержащий путь к другому файлу.

**3. Сравнение жестких и символических ссылок:**

* **Жесткая ссылка:**
  + Указывает на ту же внутреннюю структуру данных, что и исходный файл.
  + Увеличивает файловый счетчик i-узла.
  + Не может пересекать границы дисков или указывать на удаленные компьютеры.
* **Символическая ссылка:**
  + Содержит имя другого файла.
  + Не увеличивает файловый счетчик i-узла.
  + Может пересекать границы дисков и указывать на удаленные компьютеры.

**4. Дополнительные сведения:**

* В тексте также приводятся примеры системных вызовов для работы с каталогами в UNIX.
* Описаны преимущества и недостатки жестких и символических ссылок.

**7. Особые обозначения:**

* /usr/ast/mailbox – абсолютное имя файла
* mailbox – относительное имя файла
* .. – родительский каталог
* cp ../lib/dictionary . – команда копирования файла

**Реализация файловой системы**

**1. Структура файловой системы:**

* Файловая система организует данные на диске.
* Она состоит из разделов, каждый из которых содержит свою файловую систему.
* Раздел содержит загрузочный блок, суперблок, информацию о свободном пространстве, i-узлы, каталоги и файлы.

**2. Реализация файлов:**

* **Непрерывное размещение:**
  + Простая схема, но приводит к фрагментации диска.
  + Используется для компакт-дисков.
* **Связанный список:**
  + Эффективное использование дискового пространства, но медленный произвольный доступ.
* **Связанный список с таблицей в памяти (FAT):**
  + Улучшает произвольный доступ, но требует много оперативной памяти.
* **I-узлы:**
  + Гибкая схема, хорошо подходит для больших дисков.

**3. Реализация каталогов:**

* Каталоги хранят информацию о файлах, включая имя, атрибуты и местоположение на диске.
* Записи каталога могут содержать фиксированную информацию или ссылку на i-узел.
* Для ускорения поиска используются хэш-таблицы и кэширование.

**4. Примеры:**

* NTFS (Windows) использует i-узлы.
* EXT4 (Linux) использует i-узлы.
* FAT32 используется на USB-накопителях.