

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Часть 2

Учебно-методическое пособие для вузов

Составитель
М.К. Чернышов

Воронеж
Издательский дом ВГУ
2014

ЛЕКЦИЯ 8

ФАЙЛОВЫЕ СИСТЕМЫ

ВВЕДЕНИЕ

Большинство пользователей компьютеров знакомы с концепцией файла (file) – именованного набора данных, с которым можно выполнять различные операции. Файлы обычно размещаются на внешних накопителях, например, жестких дисках, оптических накопителях и лентах, хотя файлы могут существовать и непосредственно в энергозависимой оперативной памяти. В этой главе мы изучим, как системы организуют данные в файлах и обращаются к ним таким образом, чтобы обеспечить быстрое извлечение данных с накопителей, обладающих большими задержками. Мы также узнаем, как операционные системы могут создавать интерфейсы, дающие возможность пользователям выполнять навигацию в файловых системах. Поскольку внешние накопители часто содержат информацию, важную для нескольких пользователей, мы рассмотрим способы контроля доступа к данным в файлах. Во многих системах в файлах хранятся важные данные – о счетах, платежах или имуществе. Мы обсудим, как системы могут защитить такие данные от повреждения или потери вследствие отказов накопителей или отключения питания. И, наконец, мы обсудим выгоды применения баз данных вместо файлов в системах, работающих с большими объемами совместно используемых данных.

ИЕРАРХИЯ ДАННЫХ

Информация хранится в компьютерах в соответствии с определенной **иерархией данных** (datahierarchy). Самый нижний уровень этой иерархии состоит из битов. **Сочетания битов** (bitpatterns), составляющие двоичные коды, используются для представления всех данных в компьютерных системах. В последовательности из n битов можно хранить 2^n различных сочетаний битов.

Следующий уровень иерархии – это последовательности битов фиксированной длины, например, **байты** (bytes), символы и слова. Байт в применении к устройствам хранения обычно состоит из восьми битов. **Слово** (word) – это цепочка битов, которую процессор может обработать одновременно. Например, слово состоит из четырех байтов для 32-разрядного процессора и из 8 байтов для 64-разрядного.

Символы (characters) – это значения байтов (или групп байтов), используемые для представления цифр, букв, знаков пунктуации и специальных знаков. Во многих системах символ состоит из 8 битов, поэтому в

набор данных, воспринимаемый как единица программами. Если физическая запись содержит ровно одну логическую, то говорят, что файл состоит из **неблокированных записей** (unblocked records). Если физическая запись содержит несколько логических, то говорят, что файл состоит из **сблокированных записей** (blocked records). В файле с фиксированной длиной записи все записи имеют одинаковую длину; в файле с записями произвольной длины записи могут иметь любую длину вплоть до размера блока.

ФАЙЛОВЫЕ СИСТЕМЫ

Файловая система (file system) организует файлы и управляет доступом к хранящимся в них данным. Файловая система отвечает за:

- **управление файлами** (file management) – реализует механизмы хранения файлов, обращения к ним, их разделения и обеспечения их безопасности;
- **управление вспомогательными устройствами хранения** (auxiliary storage management) – выделяет пространство под файлы на вторичных и третичных устройствах хранения;
- **целостность файлов** (file integrity) – гарантирует, что хранимая в файлах информация не будет повреждена. Если целостность файлов гарантируется, то в файлах будет только та информация, которая должна быть;
- **методы доступа** (access methods) – методы, позволяющие получить доступ к хранимым данным.

Файловая система в основном занимается управлением вторичными устройствами хранения, особенно дисковыми накопителями, но она может обращаться и к данным в файлах, хранящихся на других носителях (т.е. в основной памяти).

Файловые системы позволяют пользователям создавать, изменять и удалять файлы; они также должны предоставлять возможность структурирования файлов способами, удобными для приложений и переноса данных между файлами. Пользователи должны иметь возможность предоставлять другим пользователям доступ к своим файлам. Механизм разделения файлов (совместного доступа к файлам) должен реализовать различные типы контролируемого доступа – например, **доступ для чтения** (read access), **доступ для записи** (write access), **доступ для выполнения** (execute access) и их сочетания.

Файловые системы должны обладать **аппаратной независимостью** (device independence) – пользователи должны иметь возможность обращаться к файлам по **символьным именам** (symbolic names), а не по **физическим именам** (physical names). Символьные имена – это логические, дружественные пользователям имена, например, myDirectory:Myfile.txt.

Физические имена указывают, где расположен файл, например, на диске 2, в блоках 782–791. Символьные имена дают системе возможность реализовать для пользователей **логическое представление** (logical view) их данных, присваивая осмысленные имена файлам и операциям над файлами. **Физическое представление** (physical view) связано с размещением данных файла на физическом накопителе и специфичными для накопителя операциями работы с данными. Пользователю не нужно интересоваться физическими накопителями, на которых хранятся его данные, формой, которую эти данные принимают на накопителях или физическими методами передачи данных с накопителей или на накопители.

Для проектирования файловой системы требуется знать параметры пользователей, включая количество пользователей, среднее число и размер файлов отдельного пользователя, среднюю продолжительность сеанса работы, характер выполняемых в системе приложений и так далее. Эти факторы нужно тщательно взвесить, чтобы определить наиболее оптимальную организацию файлов и структуру директорий.

Чтобы предотвратить как случайную потерю данных, так и их намеренное уничтожение, система должна располагать возможностями **резервного копирования** (backup), позволяющими создавать избыточные копии данных, и возможностями **восстановления** (recovery), позволяющими восстанавливать потерянные или поврежденные данные. В средах, решающих важные задачи, где данные должны быть защищены от несанкционированного доступа, например, в банковских системах, системах хранения оперативно-следственных данных или медицинских системах, файловые системы могут также предоставлять возможности **шифрования** (encryption) и **дешифрования** (decryption) данных. Эти возможности делают данные доступными только для тех, для кого они предназначены – тех, кто обладает ключами к шифрам.

1. Директории

Представьте себе большую систему с разделением времени, поддерживающую множество пользователей. У каждого пользователя может быть несколько профилей; в каждом профиле может быть множество файлов. Некоторые файлы могут быть маленькими, например, сообщения электронной почты. Другие файлы могут быть большими, например, списки частей в системе учета инвентаря.

Весьма часто пользовательские профили состоят из сотен или даже тысяч файлов. Если сообщество пользователей состоит из нескольких тысяч членов, то на накопителях системы могут храниться миллионы файлов. К этим файлам нужно обращаться быстро, чтобы ограничить время реакции системы.

Поле	Описание
Имя	Символьная строка, содержащая имя файла
Местоположение	Физический блок или логический адрес файла в файловой системе (например, путь к файлу)
Размер	Количество байтов, занимаемых файлом
Тип	Описание назначения файла (например, файл данных или файл директории)
Время обращения	Время последнего обращения к файлу
Время изменения	Время последнего изменения файла
Время создания	Время создания файла

Рис. 8.1. Пример содержимого файла директории

Чтобы организованно размещать и быстро находить файлы, файловые системы используют **директории** (directories). Это файлы, содержащие списки имен файлов и информацию об их размещении в системе. В отличие от других файлов, директории не хранят пользовательские данные. На рисунке 8.1 перечислены типичные поля файла директории.

2. Одноуровневые файловые системы

Простейшая организация файловой системы – это **одноуровневая** (single-level), или **плоская** (flat), файловая система. В таких системах все файлы хранятся в одной директории. В одноуровневых системах имя каждого файла должно быть уникальным. Поскольку в большинстве сред существует большое количество файлов, многие из которых используют одинаковые имена, одноуровневые системы применяются редко.

3. Иерархически структурированные файловые системы

Для большинства сред лучше подходит файловая система, организованная следующим образом (см. рис. 8.2). **Корень** (root) отмечает, в какой позиции устройства хранения начинается **корневая директория** (root directory). Директории – это файлы, которые могут указывать на другие файлы и директории. На рисунке 8.2 корневая директория указывает на различные **пользовательские директории** (user directories). Пользовательская директория содержит запись о каждом находящемся в ней пользовательском файле; каждая такая запись указывает позицию, в которой соответствующий файл находится на устройстве хранения.