Операционные системы

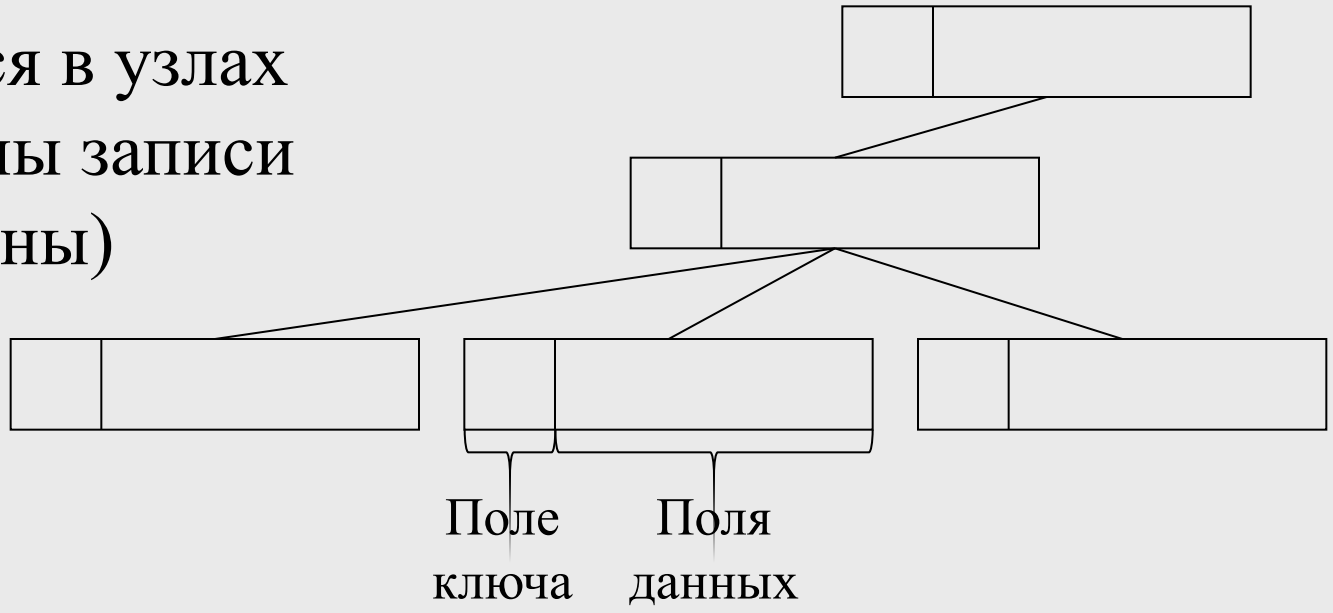
Файловые системы (часть 1)

1. Базовые методы организации ФС
   - 1.1. Общие концепции
     - 1.1.1. Структурная организация файлов
     - 1.1.2. Атрибуты файла
     - 1.1.3. Работа с файлами
     - 1.1.4. Модельная организация каталогов ФС
   - 1.2. Подходы в практической реализации ФС
     - 1.2.1. Модели реализации файлов
     - 1.2.2. Таблица размещения файловой системы
     - 1.2.3. Модели организации каталогов
     - 1.2.4. Соответствие имени и содержимого файла
     - 1.2.5. Учёт свободных блоков ФС
     - 1.2.6. Квотирование пространства ФС
   - 1.3. Надёжность файловой системы
   - 1.4. Проверка файловой системы

Структурная организация файлов

1. Файл, как **последовательность байтов**
2. Файл, как **последовательность записей переменной длины**
3. Файл, как **последовательность записей постоянной длины**
4. Иерархическая организация файлов (дерево)

Записи находятся в узлах дерева (возможны записи переменной длины)



Атрибуты файла

- Имя
- Права доступа
- Персонификация (создатель, владелец)
- Тип файла
- Размер записи
- Размер файла
- Указатель чтения / записи
- Время создания
- Время последней модификации
- Время последнего обращения
- Предельный размер файла
- …

Основные сценарии работы с файлами

Начало

Открытие файла (регистрация в системе возможности работы процесса с содержимым файла)

Работа с содержимым файла, с атрибутами файла

Завершение

Закрытие файла — информация системе о завершении работы процесса с открытым файлом

Файловый дескриптор — системная структура данных, содержащая информацию об актуальном состоянии открытого файла.

Типовые программные интерфейсы работы с файлами

open — открытие / создание файла

close — закрытие

read / write — читать, писать (относительно положения указателя чтения / запись)

delete — удалить файл из файловой системы

seek — позиционирование указателя чтение/запись

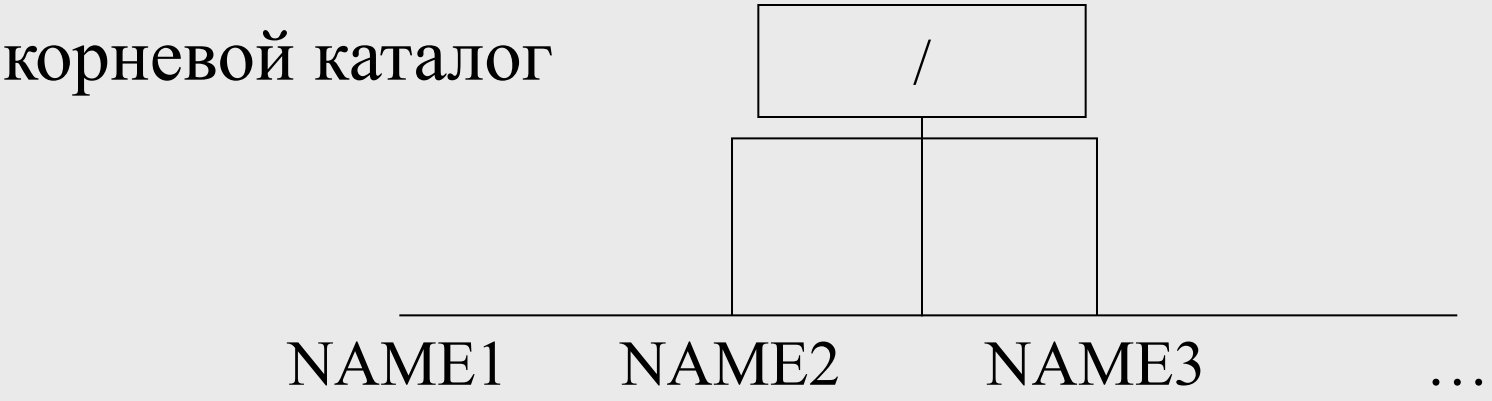
rename — переименование файла

read / write _attributes — чтение, модификация атрибутов файла

Модельная организация каталогов файловых систем

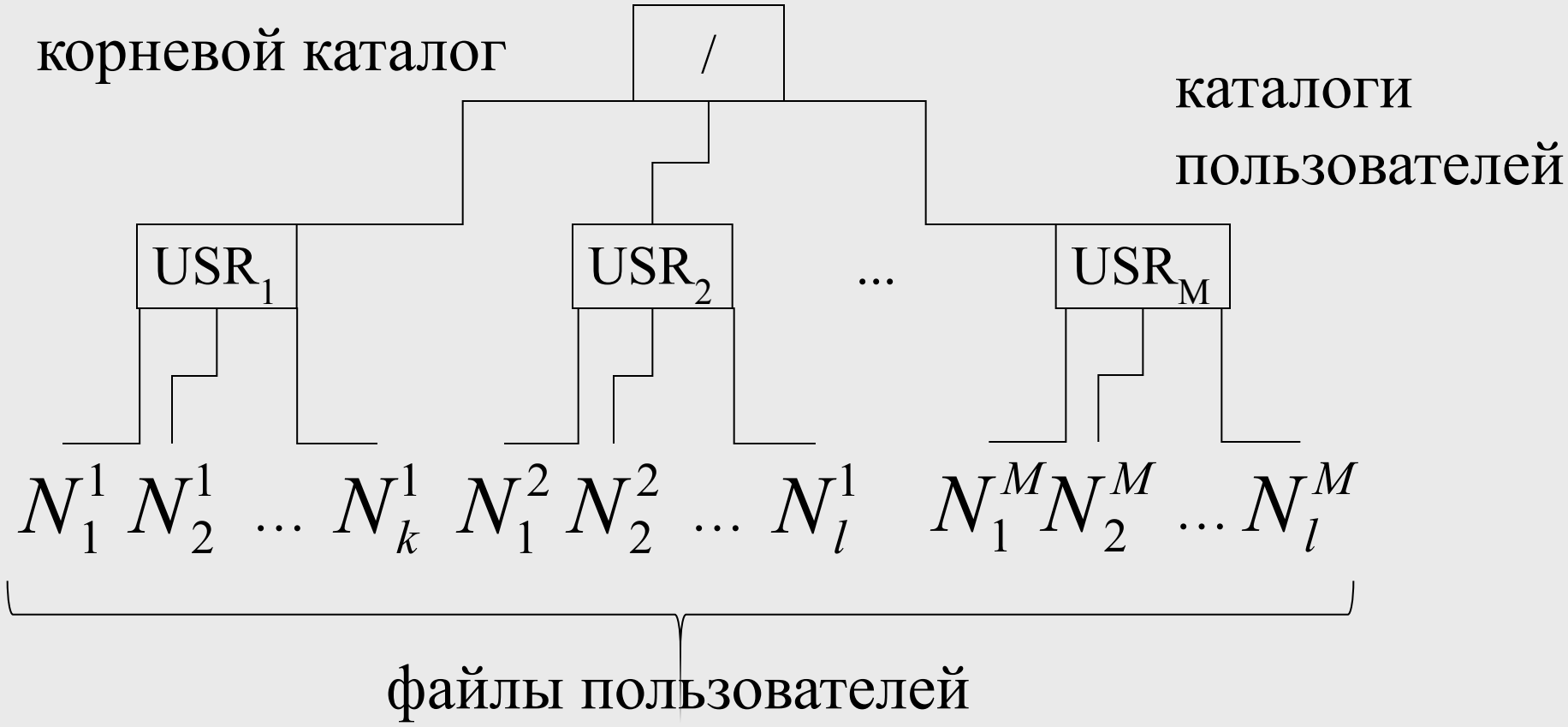
Каталог — компонент файловой системы, содержащий информацию о содержащихся в файловой системе файлах. Каталоги являются специальным видом файлов.

Модель одноуровневой файловой системы



Модельная организация каталогов файловых систем

Модель двухуровневой файловой системы

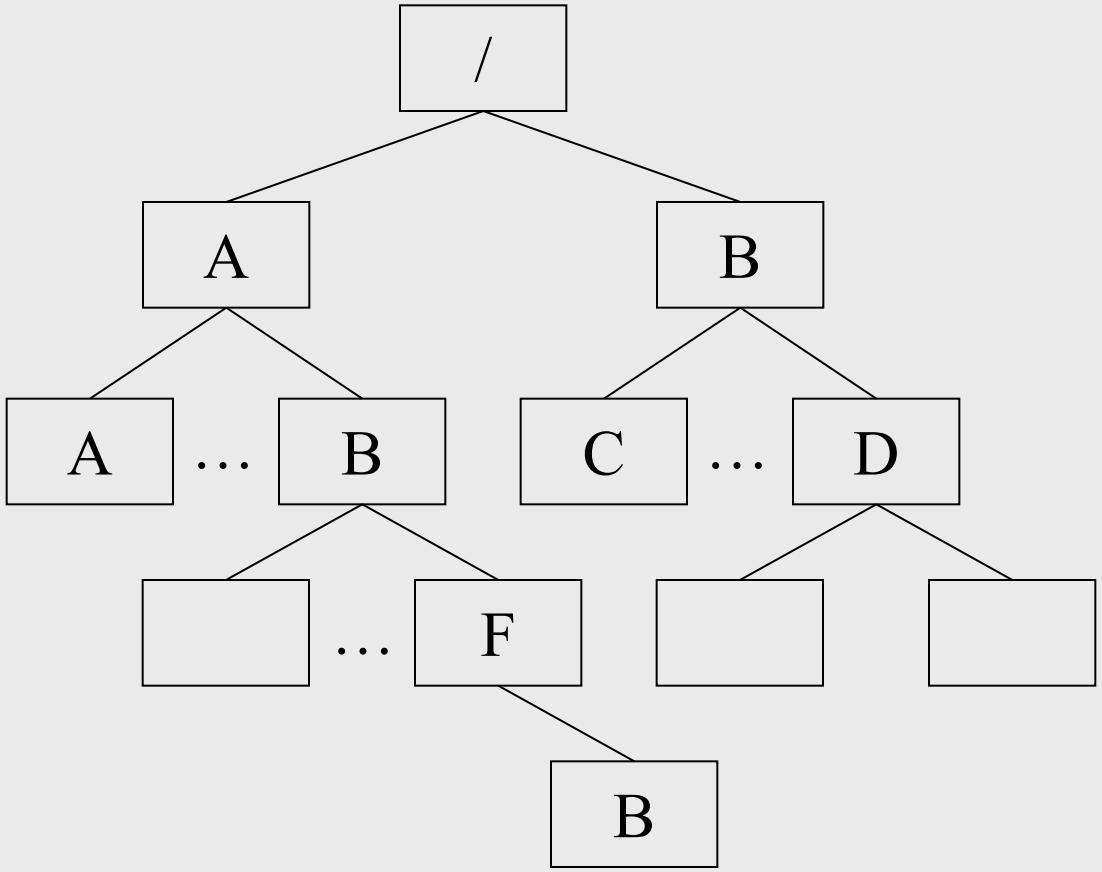


Модельная организация каталогов файловых систем

Иерархические файловые системы

Понятия

- имя файла
- полное имя файла
- относительное имя
- домашний каталог
- текущий каталог



Подходы в практической реализации файловой системы

Структура «системного» диска

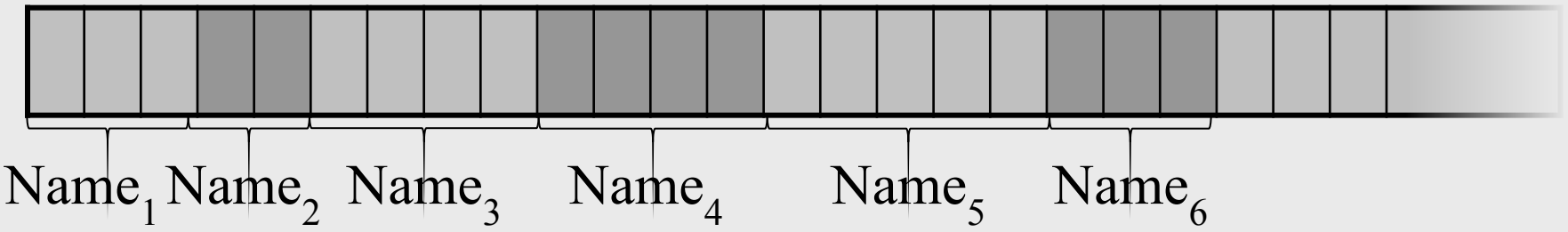
Блок физического HDD Блок виртуального HDD

Блок файловой системы

Блок файла

Модели реализации файлов

Непрерывные файлы



Достоинства

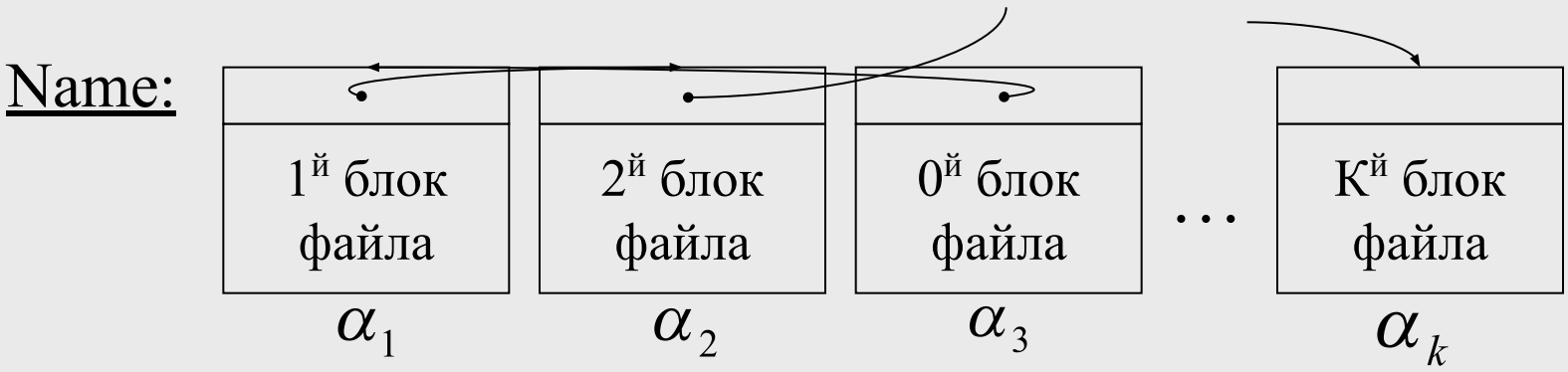
- Простота реализации
- Высокая производительность

Недостатки

- Фрагментация свободного пространства
- Возможность увеличения размера существующего файла

Модели реализации файлов

Файлы, имеющие организацию связанного списка



$\{\alpha_i\}$ — множество блоков файловой системы, в которых размещены блоки файла Name.

Достоинства

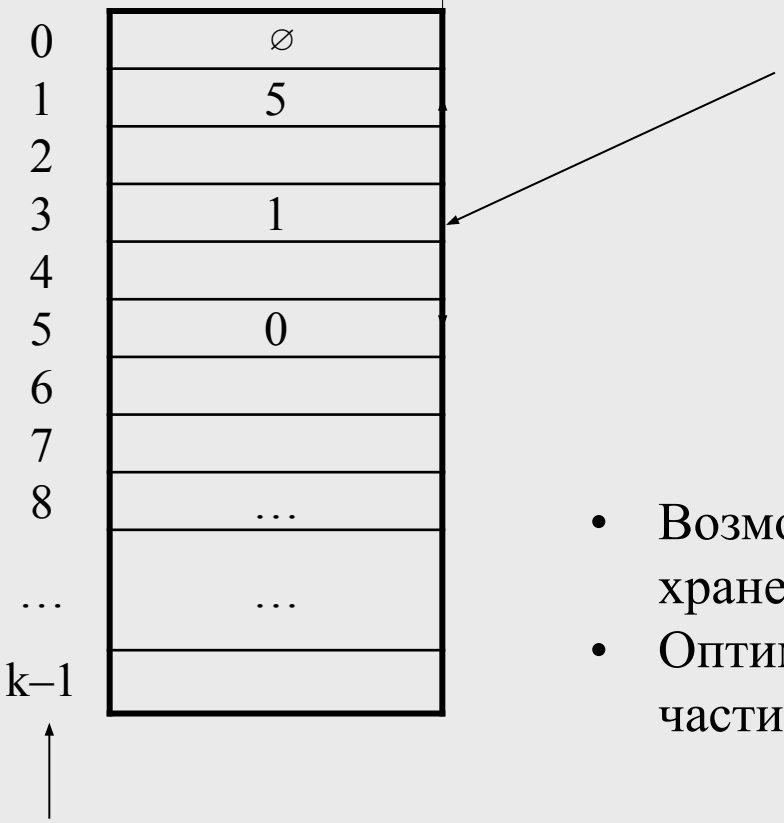
- Отсутствие фрагментации свободного пространства (за исключением блочной фрагментации)
- Простота реализации
- Эффективный последовательный доступ

Недостатки

- Сложность (неэффективность) организации прямого доступа
- Фрагментация файла по диску
- Наличие ссылки в блоке файла (ситуации чтения 2-х блоков при необходимости чтения данных объемом один блок)

Таблица размещения файловой системы

(File Allocation Table — FAT)



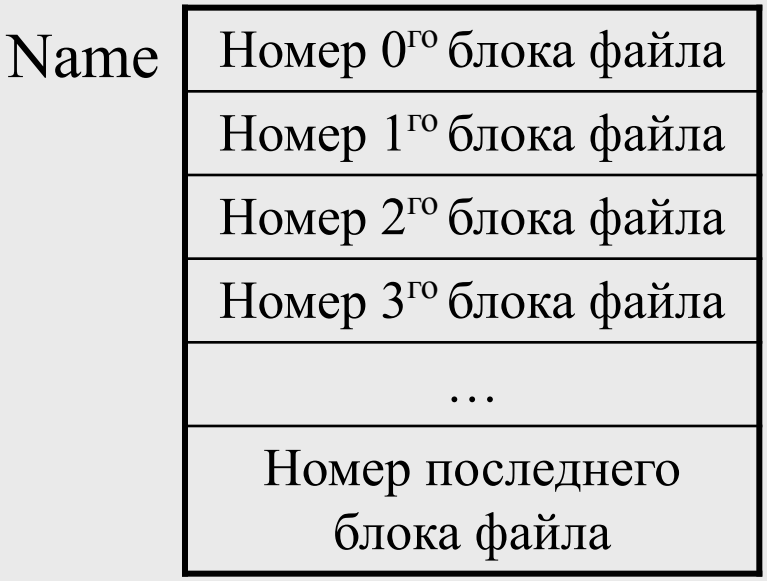
Достоинства

- Возможность использования всего блока для хранения данных файла
- Оптимизация прямого доступа (при полном или частичном размещении таблицы в ОЗУ)

Недостатки

- Желательно размещение <u>всей</u> таблицы в ОЗУ

Индексные узлы (дескрипторы)



Индексный узел (дескриптор) — системная структура данных, содержащая информацию о размещении блоков конкретного файла в файловой системе.

Достоинства

- Нет необходимости в размещении в ОЗУ информации всей FAT о всех файлах системы, в памяти размещаются атрибуты, связанные только с открытыми файлами

Недостатки

- Размер файла и размер индексного узла (в общем случае прийти к размерам таблицы размещения).

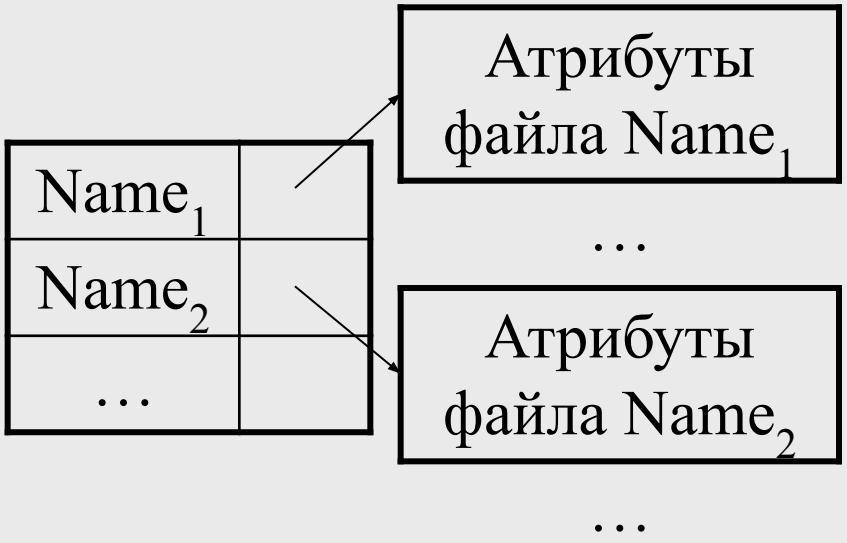
Решение: — ограничение размера файла

— иерархическая организация индексных узлов

Модели организации каталогов

$Name_1$	Атрибуты
$Name_2$	Атрибуты
…	

Записи каталога фиксированного размера, содержат имя файла и все его атрибуты.

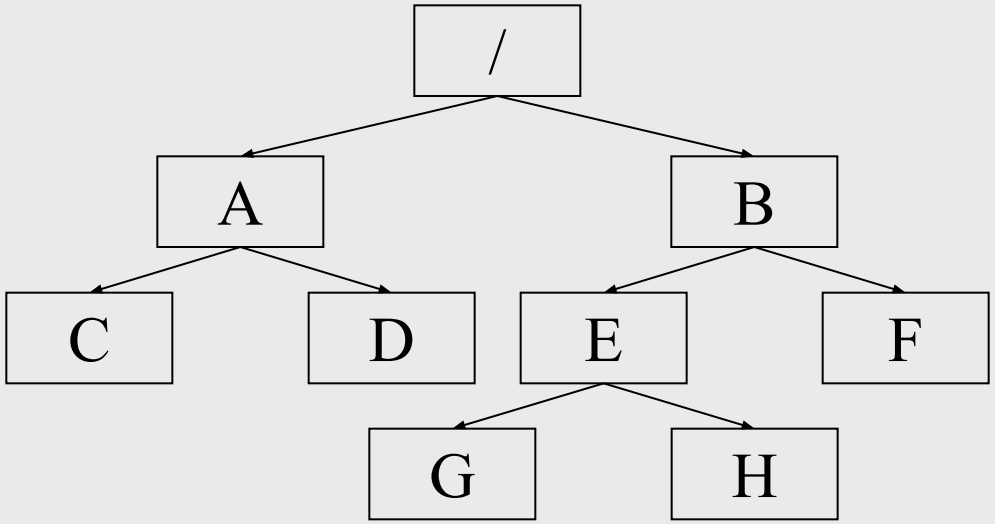


Каталог содержит имя файла и ссылку на атрибуты файла. Размер атрибутов может варьироваться.

Организация «длинных» имен файлов?

Взаимнооднозначное соответствие: имя файла — содержимое файла

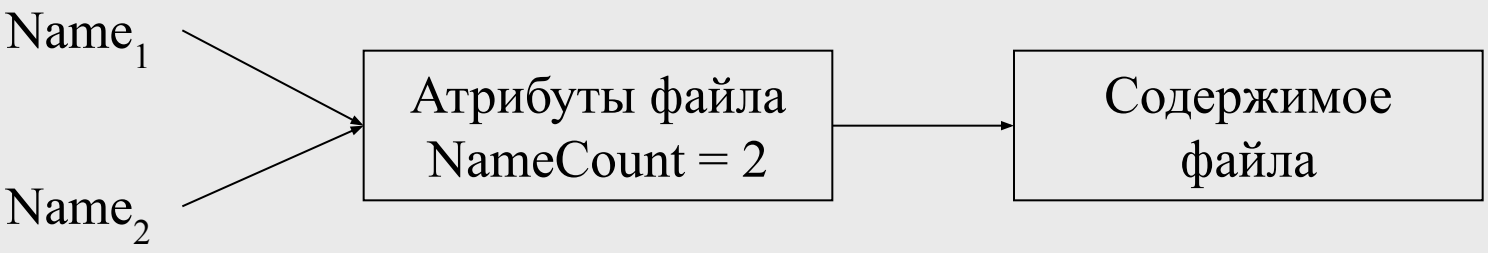
1. Содержимому любого файла соответствует единственное имя файла



Взаимнооднозначное соответствие: имя файла — содержимое файла?

2. Содержимому файла может соответствовать два и более имен файла.

   2.1 «Жесткая» связь



   2.2 “Символическая” связь

Координация использования пространства внешней памяти

Проблема — размер блока файловой системы.

«Большой блок»:

- эффективность обмена
- существенная внутренняя фрагментация (неэффективное использование пространства ВП)

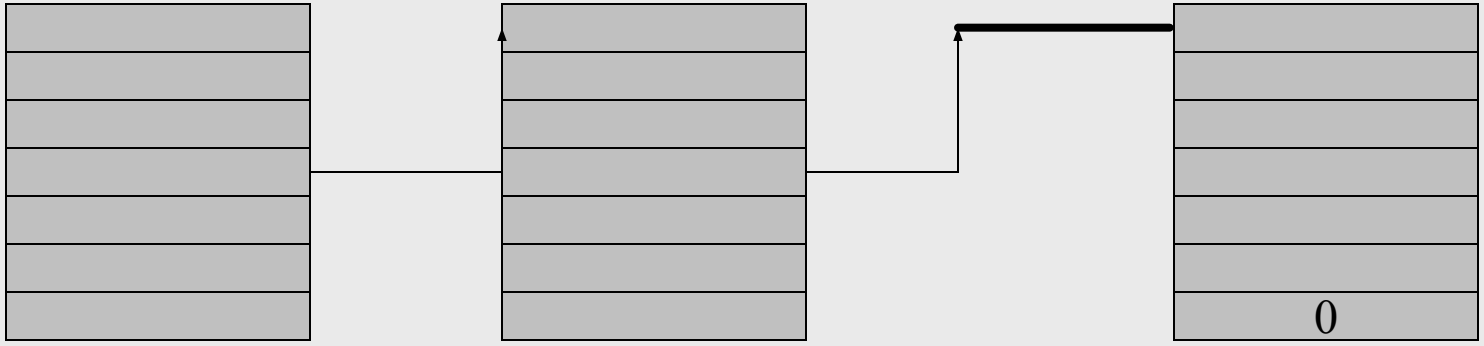
«Маленький блок»:

- эффективное использование пространства ВП
- фрагментация данных файла по диску

Проблема — определение оптимального размера блока.

Учет свободных блоков файловой системы

Связный список свободных блоков

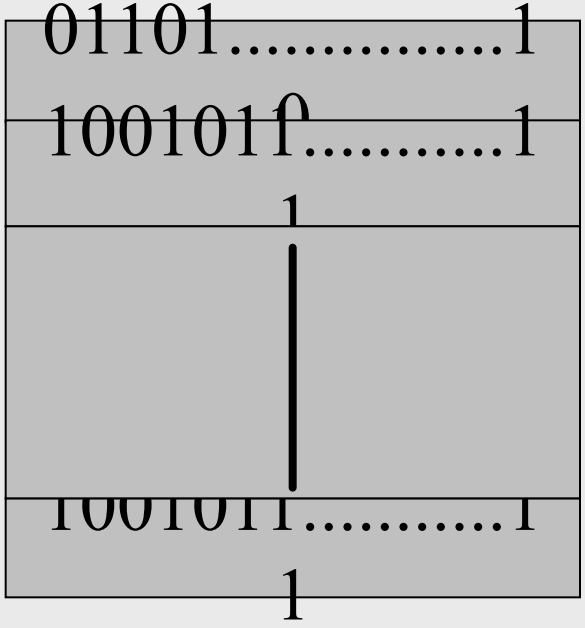


При использовании связного списка свободных блоков в ОЗУ размещается первый блок списка.

Использование битового массива

Состояние любого блока определяется содержимым бита с номером каждого блока.

Если блок свободен, бит равен 1, занят — 0.



Квотирование пространства файловой системы

Учет количества и размеров тех файлов, у которых атрибут владельца соответствует конкретному пользователю.

Жесткие лимиты не превышаются никогда. Гибкие квоты можно превышать, но после этого включается обратный счетчик предупреждений. Пока счетчик > 0, при каждой регистрации пользователя в системе от получает предупреждение, если счетчик = 0, пользователь блокируется.

Надежность файловой системы

- Потеря информации в результате аппаратного или программного сбоя
- Случайное удаление файлов

⇒ **Резервное копирование (архивирование):**

- Копируются не все файлы файловой системы (избирательность архивирования по типам файлов)
- Инкрементное архивирование (резервное копирование) — единожды создается «полная» копия, все последующие включают только обновленные файлы
- Использование компрессии при архивировании (риск потери всего архива из-за ошибки в чтении/записи сжатых данных)
- Проблема архивирования «на ходу» (во время копирования происходят изменения файлов, создание, удаление каталогов и т.д.)
- Распределенное хранение резервных копий

Надежность файловой системы

Стратегии архивирования

- **Физическая архивация**
  - «Один в один»
  - Интеллектуальная физическая архивация (копируются только использованные блоки файловой системы)
  - Проблема обработки дефектных блоков

- **Логическая архивация** — копирование файлов (а не блоков), модифицированных после заданной даты.

Проверка целостности файловой системы

Проблема — при аппаратных или программных сбоях возможна потеря информации:

- потеря модифицированных данных в «обычных» файлах
- потеря системной информации (содержимое каталогов, списков системных блоков, индексные узлы и т.д.)

Контроль целостности или непротиворечивости файловой системы.

Проверка целостности файловой системы

Контроль непротиворечивости блоков файловой системы:

Модельная стратегия контроля

1. Формируются две таблицы:
   - таблица занятых блоков
   - таблица свободных блоков

(размеры таблиц соответствуют размеру файловой системы — число записей равно числу блоков ФС)

Изначально все записи таблиц обнуляются.

2. Анализируется список свободных блоков. Для каждого номера свободного блока увеличивается на 1 соответствующая ему запись в таблице свободных
3. Анализируются все индексные узлы. Для каждого блока, встретившегося в индексном узле, увеличивается его счетчик на 1 в таблице занятых блоков
4. Анализ содержимого таблиц и коррекция ситуаций

Проверка целостности файловой системы

Варианты анализа таблиц

1.

0	1	2	3	4	5
1	1	0	1	0	1

Таблица занятых блоков

0	0	1	0	1	0

Таблица свободных блоков

Непротиворечивость файловой системы соблюдена.

Проверка целостности файловой системы

2.

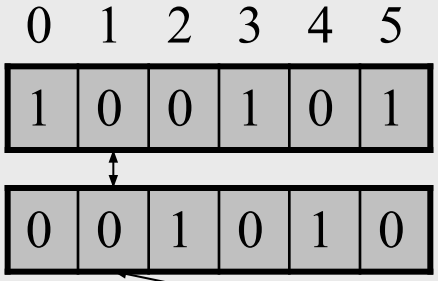


Таблица занятых блоков

Таблица свободных блоков

Пропавший блок

Оставить как есть, но система «замусоривается».

Добавить в список свободных блоков файловой системы.

3.

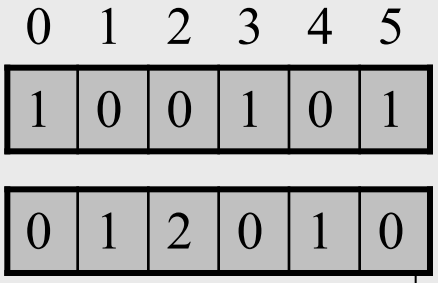


Таблица занятых блоков

Таблица свободных блоков

Дубликат свободного блока – пересоздание списка свободных блоков.

Проверка целостности файловой системы

4.

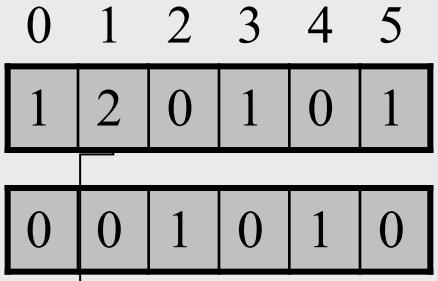


Таблица занятых блоков

Таблица свободных блоков

Дубликат занятого блока ⇒ автоматическое решение максимально затруднено, имеет место потеря информации в одном из файлов.

Действие

1. $Name_1$ → копируется $Name_1^2$
2. $Name_2$ → копируется $Name_2^2$
3. Удаляются $Name_1$, $Name_2$
4. Запускается переопределение списка свободных блоков
5. Обратное переименование файлов и фиксация факта их возможной проблемности

Проверка целостности файловой системы

Контроль непротиворечивости файлов файловой системы

Возможны варианты:

1. L = M — все в порядке
2. L > M $\Rightarrow$ NameCount = L
3. L < M