Системы файлов

Лекция 10

Операционные системы. Linux

Интерфейс с системой файлов

- **Понятие файла**
- **Методы доступа**
- **Структура директорий**
- **Монтирование файловых систем**
- **Общий доступ к файлам**
- **Защита**

Понятие файла

- **Файл (file)** – это смежная область логического адресного пространства
- **Типы:**
  - **Данные**
    - **числовые**
    - **символьные**
    - **двоичные**
  - **Программа**

Структура файла

- Последовательность слов или байтов (UNIX)
- Простая структура - последовательность записей
  - Строки
  - Записи фиксированной длины
  - Записи переменной длины
- Сложная структура
  - Отформатированный документ
  - Загрузочный модуль; PE-файл (.NET), class-файл (Java)
- Сложная структура может быть смоделирована записями путем добавления соответствующих управляющих символов.
- Интерпретируют файлы:
  - Операционная система
  - Программа

Атрибуты файлов

- **Имя (Name)** – информация в символьной форме, воспринимаемая человеком.
- **Тип (Type)** – необходим для систем, которые поддерживают различные типы файлов (Эльбрус: тип файла – *число*; 0 – данные, 2 – код, 3 – текст и т. д.). В MS DOS, Windows, UNIX тип файла принято кодировать расширением имени.
- **Размещение (Location)** – указатель на размещение файла на устройстве.
- **Размер (Size)** – текущий размер файла.
- **Защита (Protection)** – управляющая информация о том, кто может читать, изменять и исполнять файл.
- Время, дата, идентификация пользователя.
- Информация о файлах хранится в структуре директорий.

Операции над файлом

- *Создание* - Create
- *Запись* - Write
- *Чтение* - Read
- *Поиск позиции внутри файла* - Seek
- *Удаление* - Delete
- *Сокращение* - Truncate
- *Open(F_i)* – поиск в структуре директорий на диске элемента F_i, и перемещение содержимого элемента в память.
- *Close (F_i)* – переместить содержимое элемента F_i из памяти в структуру директорий на диске.

Типы файлов – имена и расширения

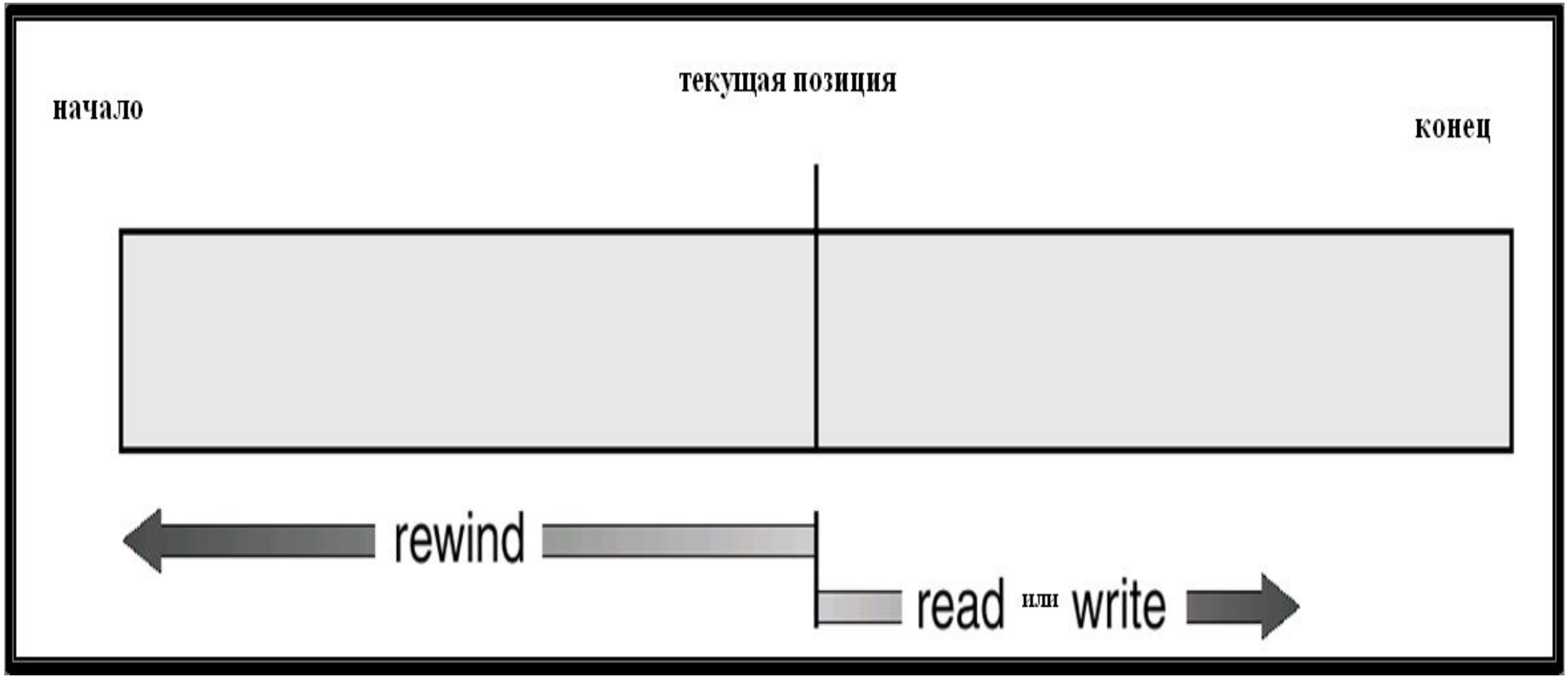
тип файла	расширение имени	функциональность
исполняемый код (загрузочный модуль)	exe, com, bin или отсутствует	готовая к выполнению программа в бинарном машинном коде
объектный модуль	obj, o	откомпилированная программа в бинарном коде, но не слинкованная
исходный код на языке программирования	c, cc, java, pas, asm, a	исходный код на различных языках (Си, Паскаль и др.)
командный файл	bat, sh	файл с командами для командного интерпретатора
текст	txt, doc	текстовые данные, документы
документ для текстового процессора	wp, tex, rrf, doc	документ в формате какого-либо текстового процессора
библиотека	lib, a, so, dll, mpeg, mov, rm	библиотеки модулей для программирования
файл для печати или визуализации	arc, zip, tar	ASCII- или бинарный файл в формате для печати или визуализации
архив	arc, zip, tar	несколько файлов, сгруппированных в один файл, для архивации или хранения
мультимедиа	mpeg, mov, rm	бинарный файл, содержащий аудио- или видео/аудиоинформацию

Методы доступа к файлам

- **Последовательный доступ**
  - *read next*
  - *write next*
  - *reset*
  - *rewrite*
- **Прямой доступ**
  - *read n*
  - *write n*
  - *position to n*
  - *read next*
  - *write next*
  - *rewrite n*

n = относительный номер блока

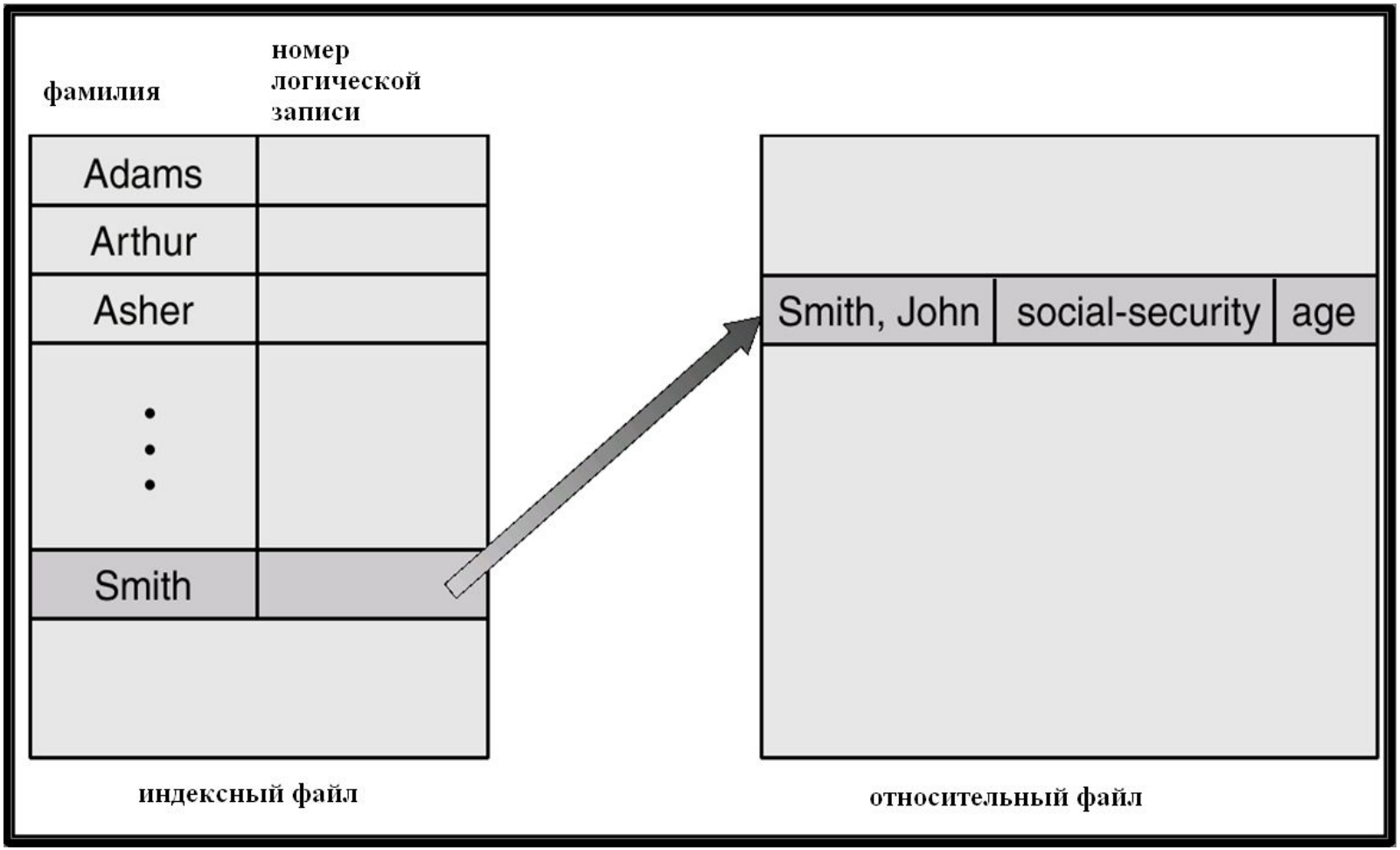
Файл последовательного доступа



Моделирование последовательного доступа для файла с прямым доступом

последовательный доступ	реализация для файла с прямым доступом
reset	*cp* = 0;
read next	*read cp*; *cp* = *cp*+1;
write next	*write cp*; *cp* = *cp*+1;

Пример индексного файла и относительного файла



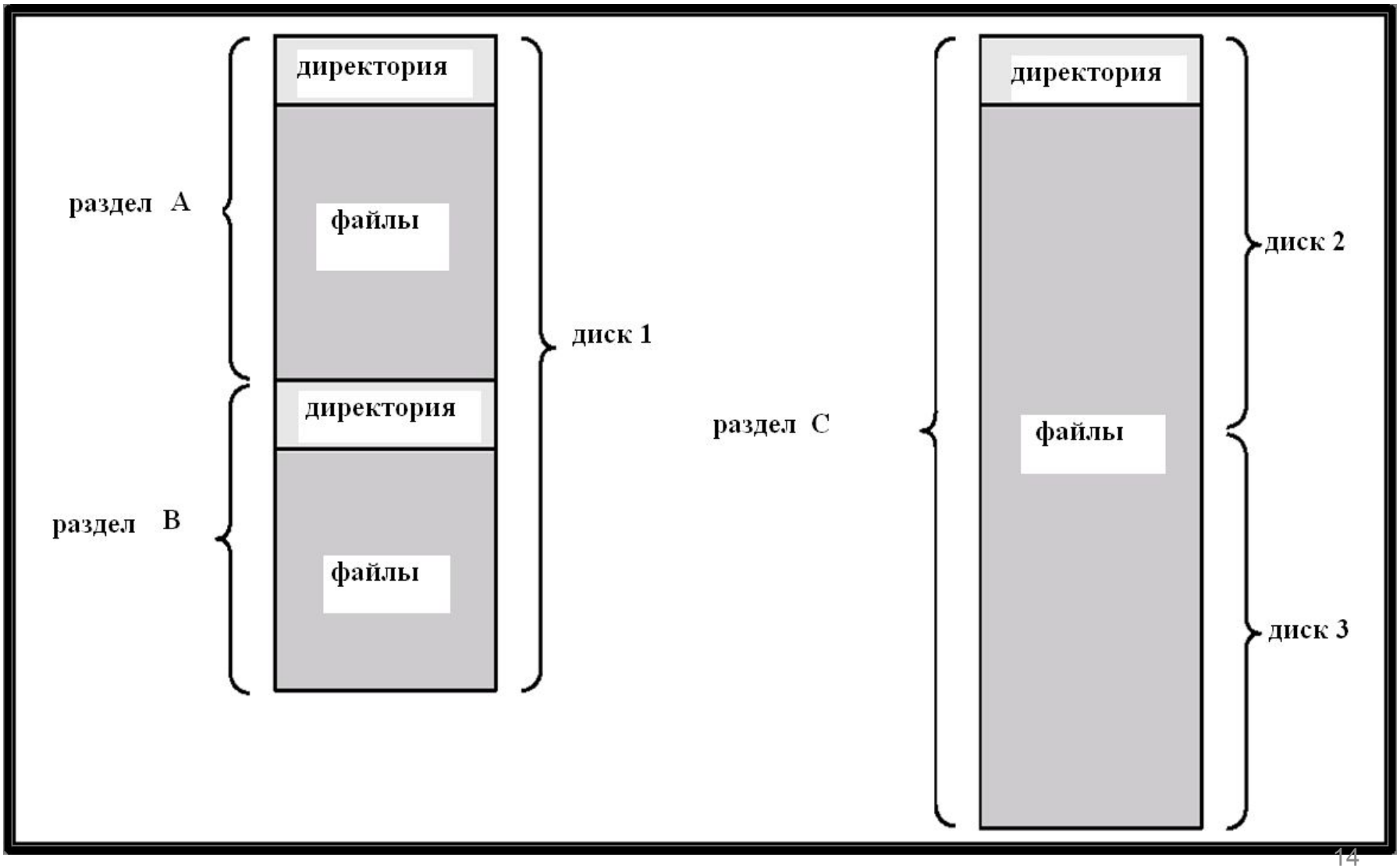
Структура директорий

- Совокупность узлов, содержащих информацию обо всех файлах.
- Как структура директорий, так и файлы хранятся на диске.
- Резервное копирование (back-up) обеих этих структур выполняется на ленту (стример), flash-память, внешний жесткий диск, CD и др.

Файловая система в “Эльбрусе”

- Файлы, контейнеры, справочники
- Файл – заголовок и память.
- В заголовке – порядка 200 (!) атрибутов файла
- Возможно создание файла и управление им без присваивания ему имени, т.е. без отображения в справочниках (директориях)
- Файлы могут ссылаться друг на друга (по файловой ссылке, а не по имени) через справочники внешних ссылок (СВС)
- *Недостатки*: сложная структура файлов, большое число атрибутов, зависимость логической структуры файла от типа устройства

Типичная организация файловой системы



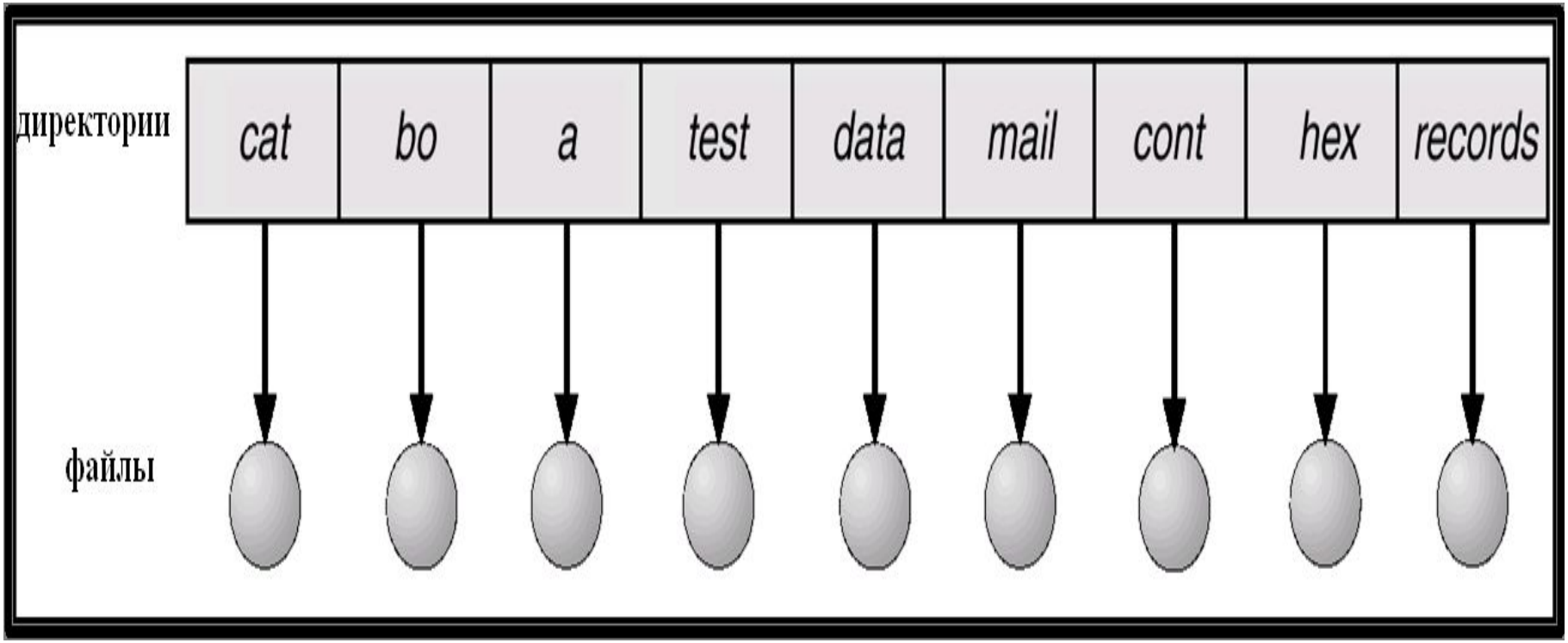
Операции над директорией

- Поиск файла
- Создание файла
- Удаление файла
- Создание поддиректории
- Вывод содержимого директории
- Переименование файла
- Создание символической ссылки
- Обход файловой системы (traverse – обход дерева)
- Ср. с “Эльбрусом”: в нем создание файла – отдельная операция, не связанная с директорией (справочником) вообще

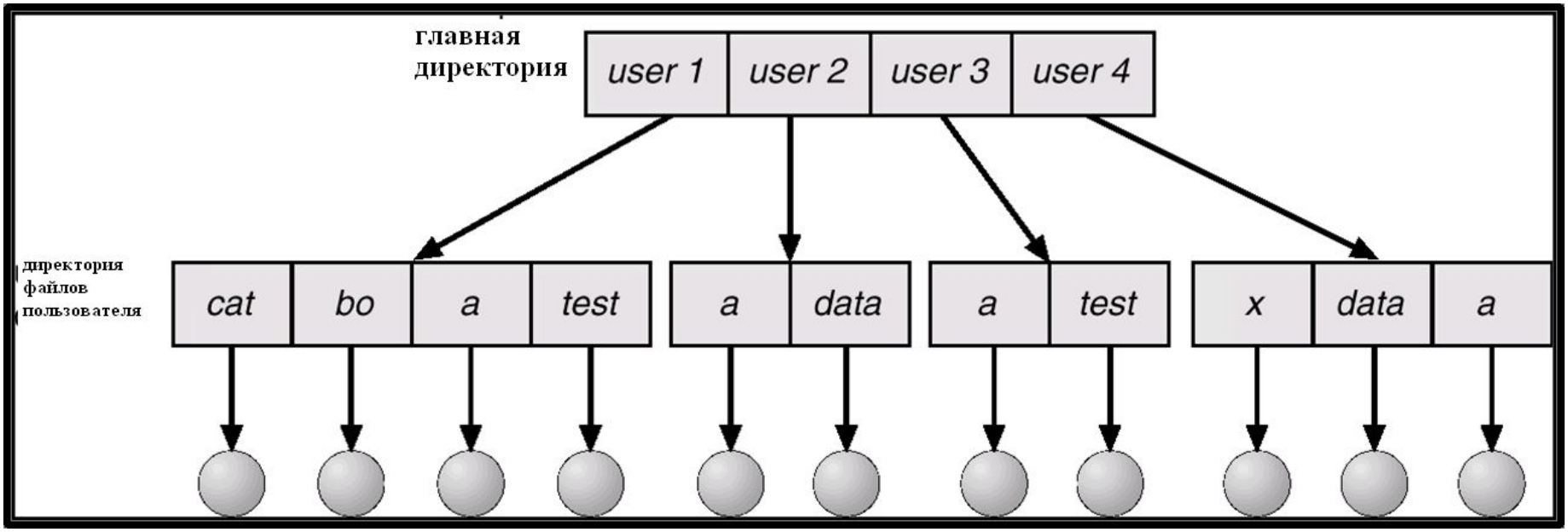
Цели логической организации директорий

- Эффективность – быстрый поиск файла.
- Именование – удобство для пользователей.
  - Два пользователя могут называть два разных файла одним и тем же именем.
  - Один и тот же файл может иметь несколько различных имен.
- Логическая группировка файлов по назначению, свойствам и т.д. (почта, Java-код, игры и др.)

Одноуровневая организация для всех пользователей – проблемы с группировкой и именованием

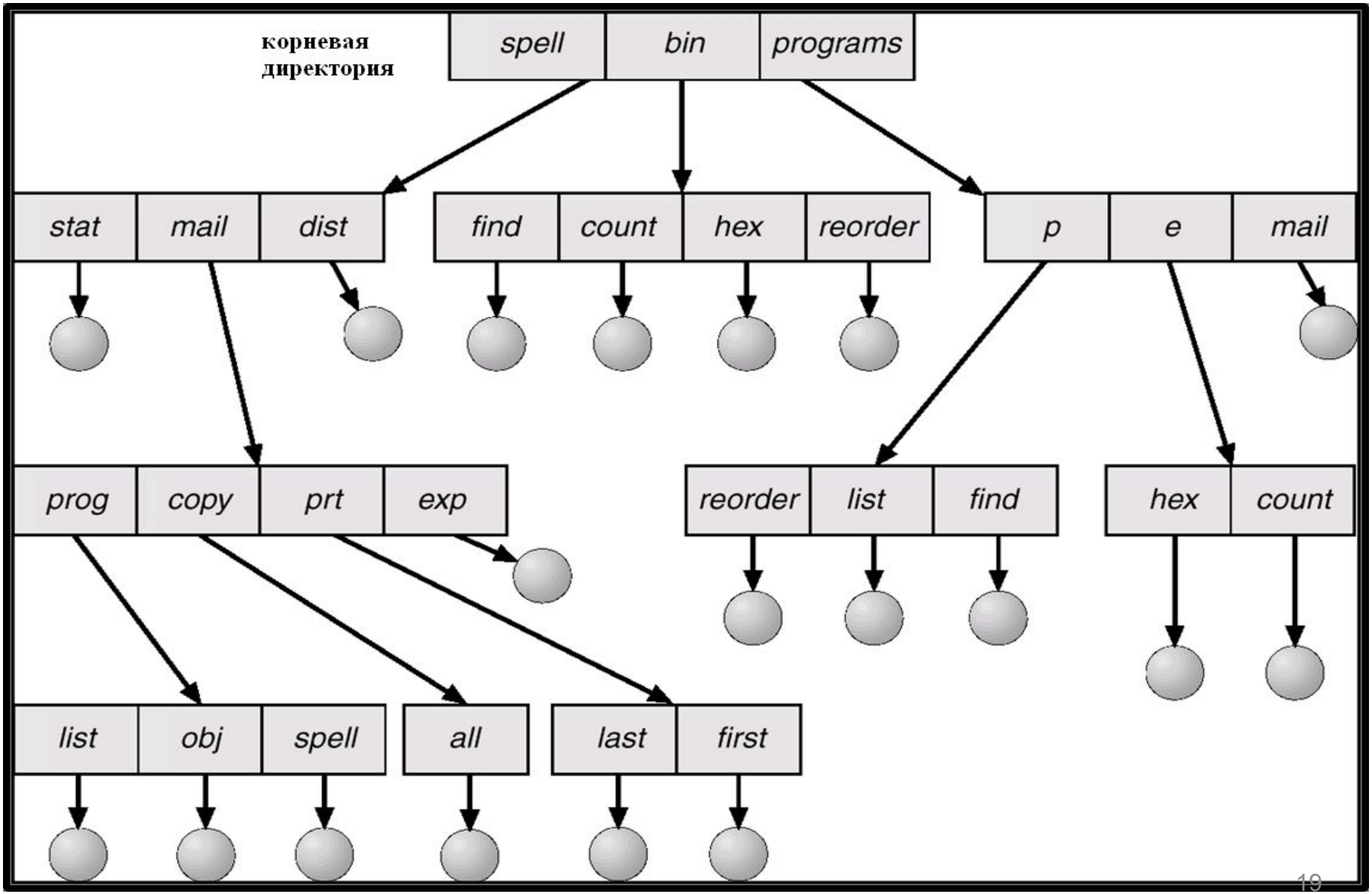


Двухуровневая организация



- Имя пути
- Возможность иметь одинаковые имена файлов для различных пользователей
- Эффективный поиск
- Нет возможности группировки

Древовидная структура директорий



Древовидная структура директорий

- Эффективный поиск.
- Возможность группировки.
- Текущая директория (рабочая директория)
  - cd /spell/mail/prog
  - type list

Древовидная структура директорий

- Абсолютный или относительный путь (первый может создавать проблемы при переносе на другие компьютеры, сборке и др.)
- Создание нового файла выполняется в текущей директории.
- Удаление файла

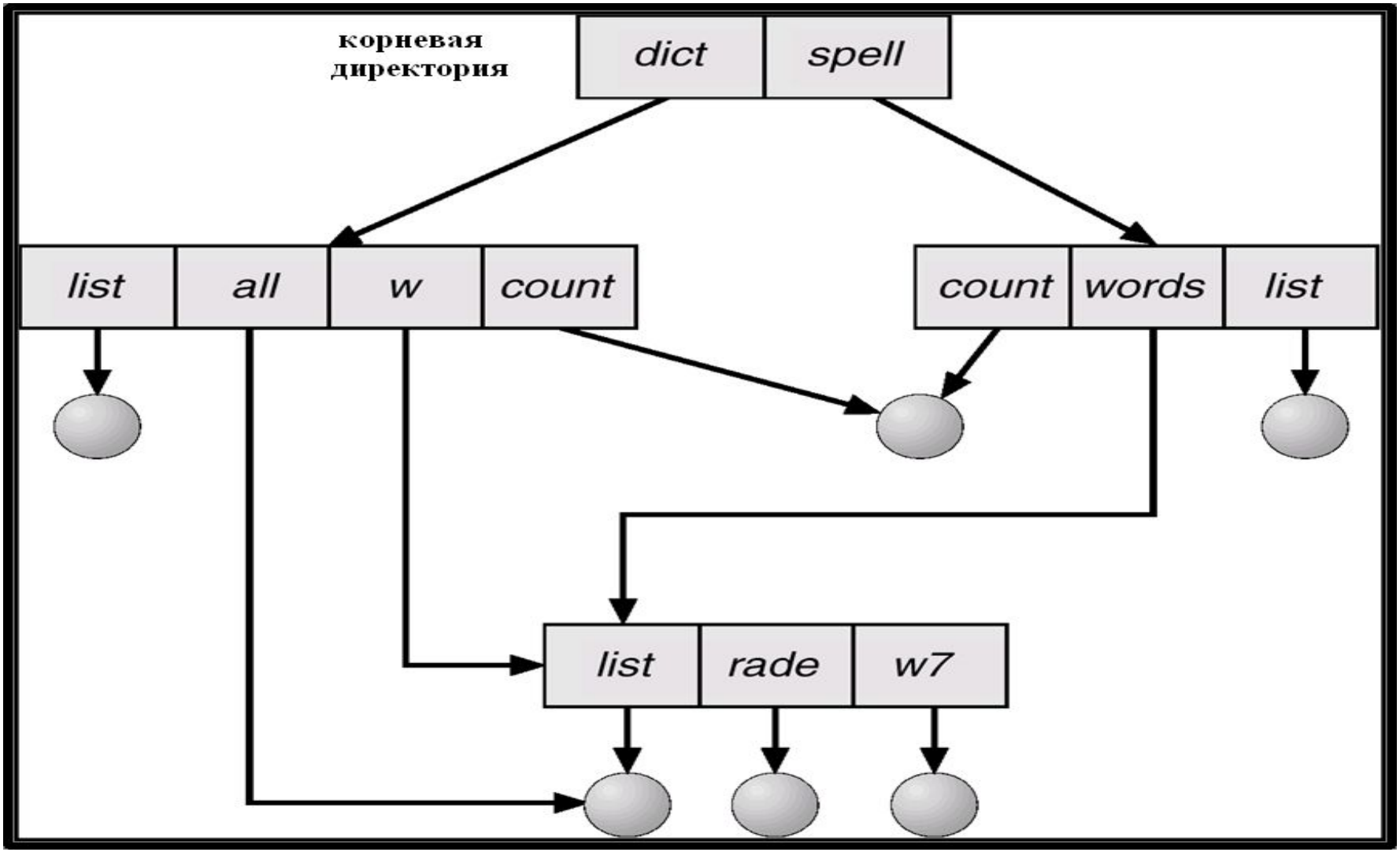
  rm <file-name>
- Создание новой поддиректории выполняется в текущей директории.

  mkdir <dir-name>

  Пример: если текущая директория /mail

  mkdir count

Структура директорий в виде ациклического графа (с разделяемыми директориями и файлами)

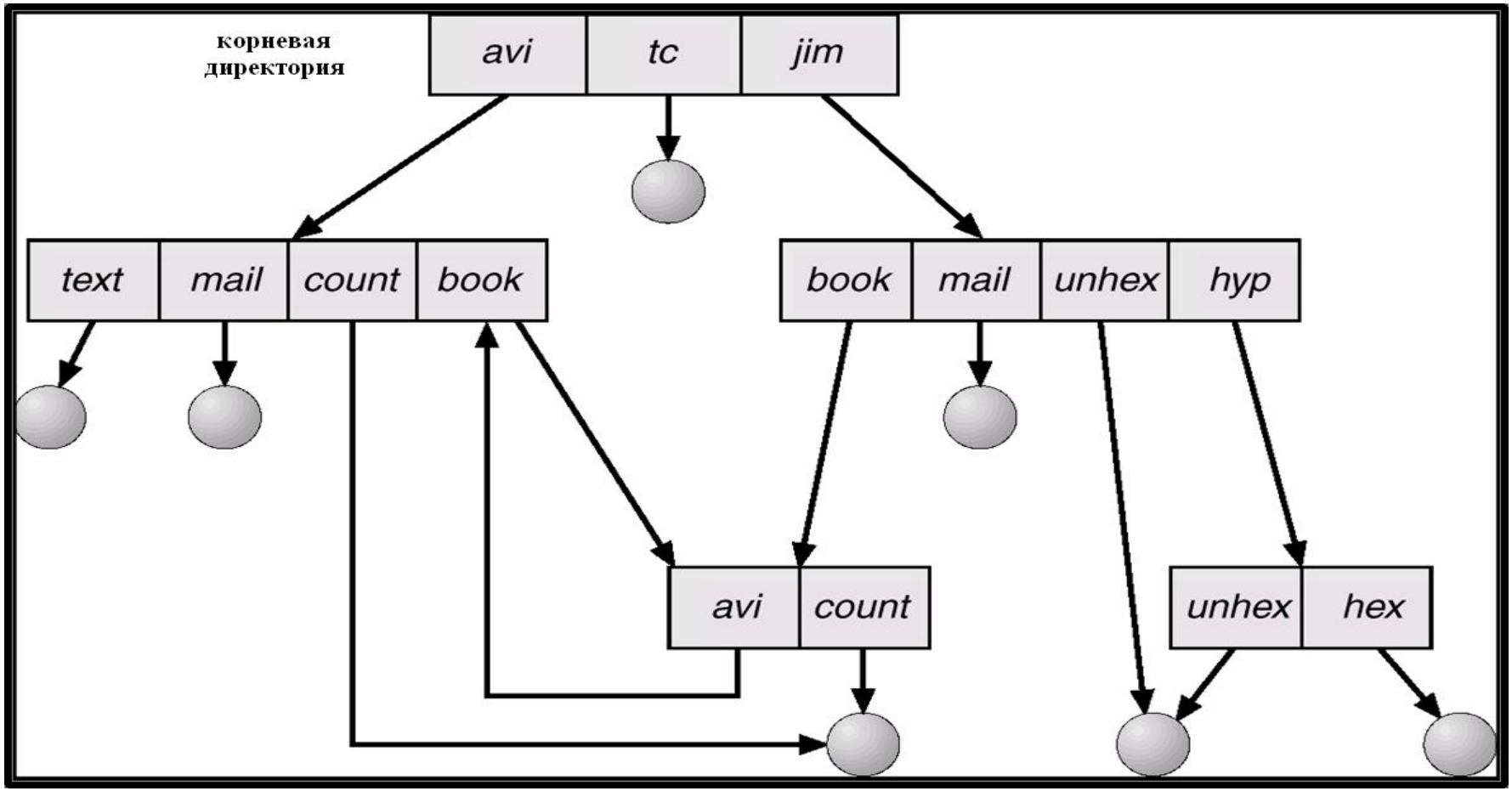


Структура директорий в виде ациклического графа (продолжение)

- Два различных имени у одного файла (директории) - aliasing
- Если в *dict* удалить *list* ⇒ подвисшая ссылка (dangling pointer).

  Решения:
  - Обратные ссылки (с целью удаления всех ссылок).
  - Счетчики ссылок.

Структура директорий в виде произвольного графа



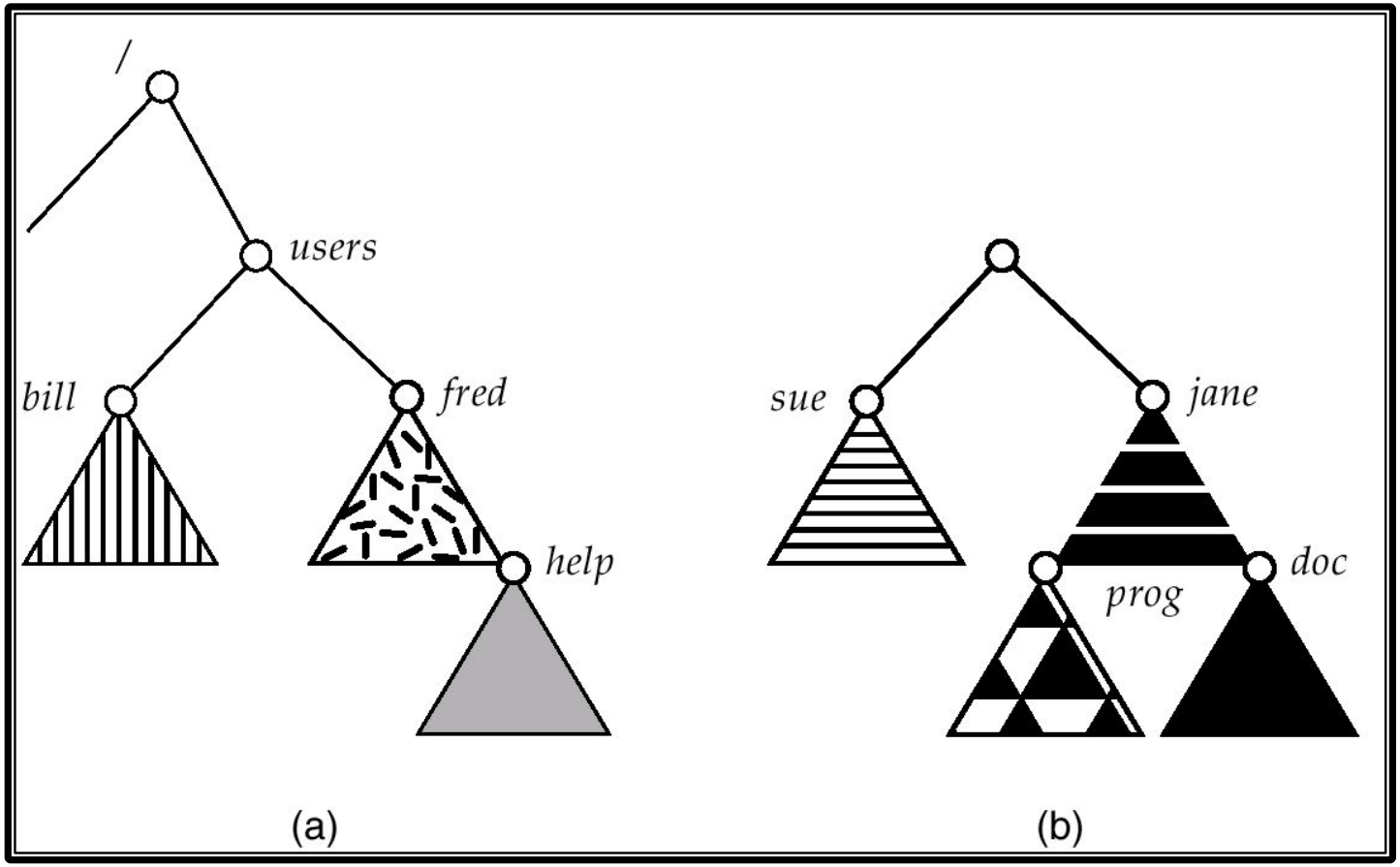
Структура директорий в виде произвольного графа (прод.)

- Как гарантировать отсутствие циклов?
  - Допускать только ссылки на файлы, а не на поддиректории.
  - Сборка мусора.
  - Каждый раз при создании новой ссылки запускать алгоритм проверки отсутствия циклов.

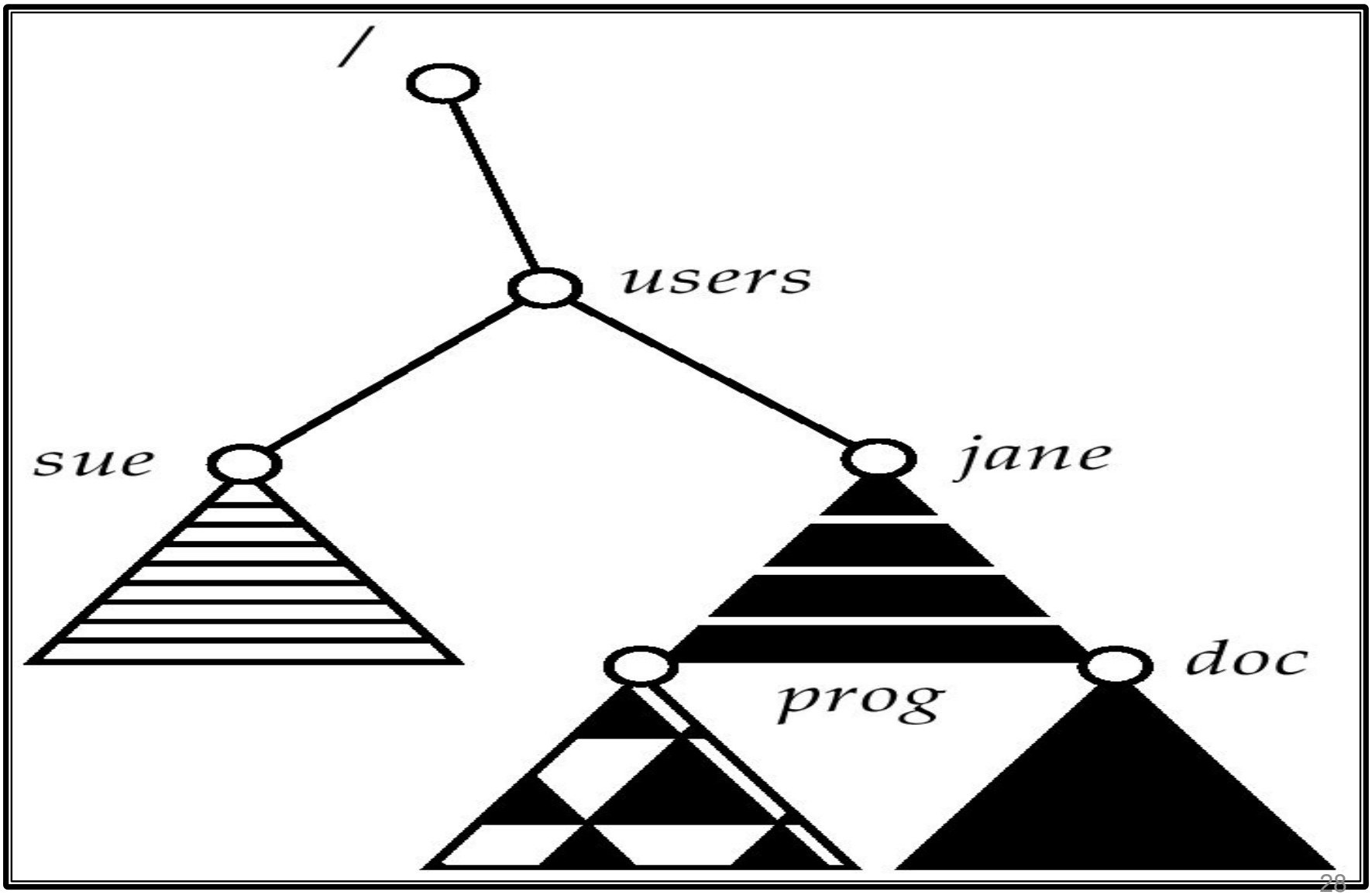
Монтирование файловых систем (mounting)

- Файловая система должна быть смонтирована, прежде чем к ней будет обеспечен доступ.
- Файловая система монтируется в некоторую точку монтирования (mount point).
- Монтирование (с абстрактной точки зрения) – подсоединение отдельного дерева (еще не смонтированной файловой системы) к какой-либо вершине (точке монтирования) общего дерева смонтированных и доступных файловых систем
- UNIX: команды *mount*; *automount*; *autodirect*

Дерево смонтированных систем (a) и еще не смонтированная файловая система (b)



Точка монтирования



Общий доступ к файлам (sharing)

- В многопользовательских системах общий доступ к файлам необходим.
- Общий доступ может быть обеспечен через некоторую систему защиты (protection).
- В распределенных системах файлы могут использоваться совместно через сеть.
- Network File System (NFS) – распространенный метод общего доступа к файлам.
- Solaris: */net/hostname/filesystem*; *hostname* – имя машины; *filesystem* – имя файловой системы на ней, для которой разрешен общий доступ (выполнена команда share)

Защита (protection)

- Создатель файла должен иметь возможность управлять:
  - Списком допустимых операций над файлом
  - Списком пользователей, которым они разрешены
- Типы доступа:
  - Read
  - Write
  - Execute
  - Append
  - Delete
  - List

Списки доступа и группы (UNIX)

- Режимы доступа: **read, write, execute**
- Три класса пользователей:

RWX

a) owner access 7 ⇒ 1 1 1

RWX

b) group access 6 ⇒ 1 1 0

RWX

c) public access 1 ⇒ 0 0 1

- Системный администратор создает группу (например, JAVA) и включает в нее нескольких пользователей.
- Для конкретного файла (например, *game*) или поддиректории определяются соответствующие полномочия доступа
- Для директории “X” означает возможность входа в нее (*cd)*

Реализация файловых систем

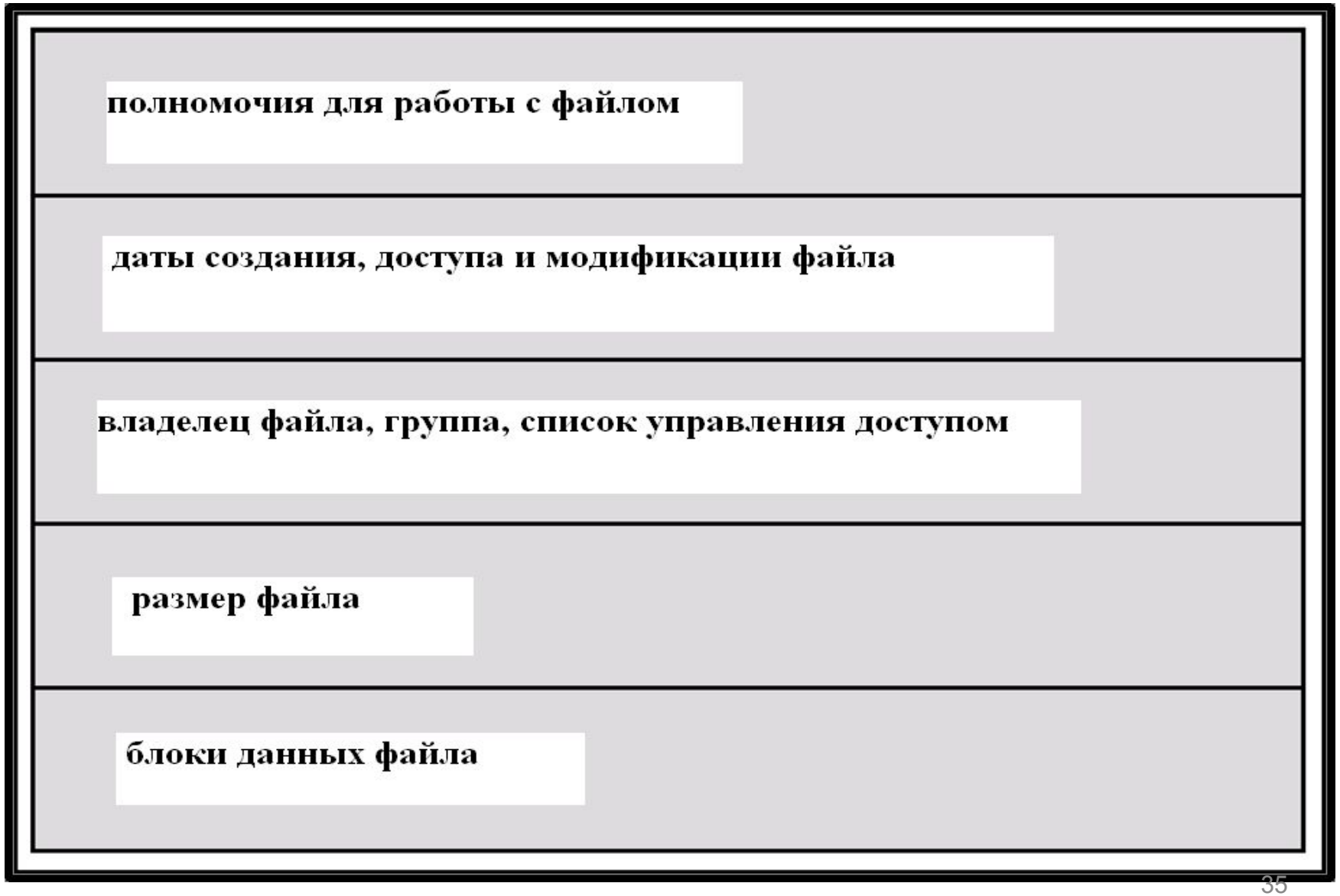
- Структура файловых систем
- Реализация файловых систем
- Реализация директорий
- Методы размещения файлов
- Управление свободной памятью
- Эффективность и производительность
- Восстановление
- Файловые системы на основе журналов транзакций (Log-Structured)
- NFS

Структура файловой системы

- Структура файла
  - **Файл** - логическая единица распределения памяти
  - **Файл** – совокупность взаимосвязанной информации
- Файловая система располагается во внешней памяти (на дисках)
- Файловая система организована по уровням.
- *Блок управления файлом (File control block)* – структура в памяти, содержащая информацию о файле.

Многоуровневая файловая система

Типовая структура блока управления файлом



Структуры ОС в памяти для управления файловой системой (открытие, чтение)