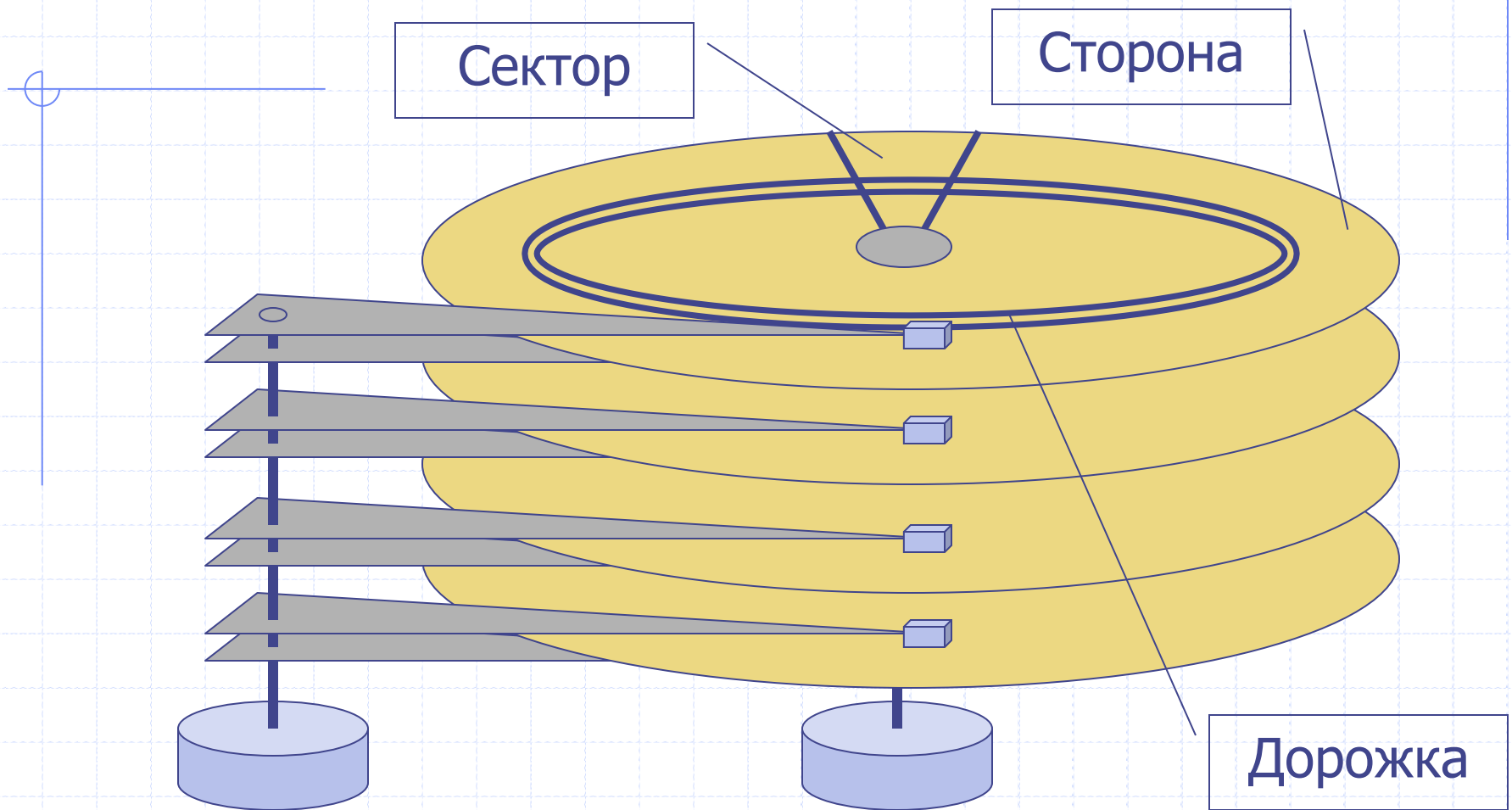


Файловые системы



Схема жесткого диска



(cylinder,head,sector) → logical sector / LBA sector

(дорожка,сторона,сектор) → логический сектор

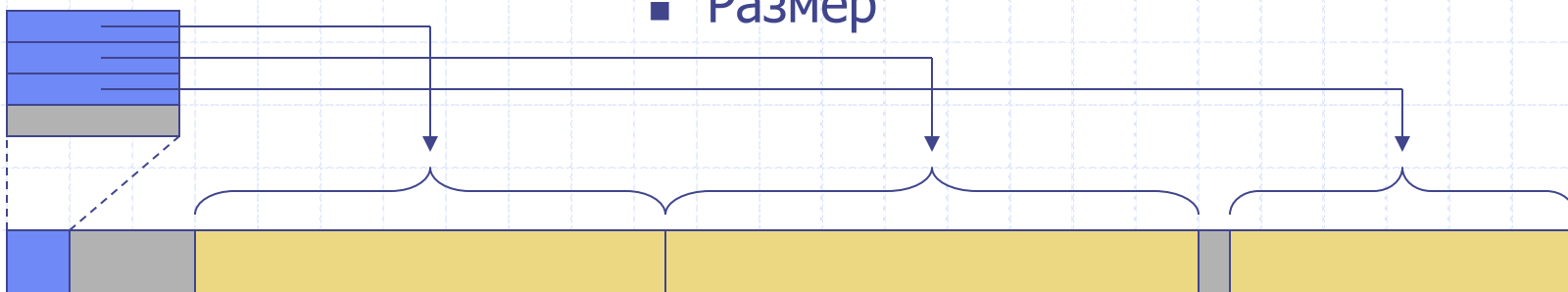
Разделение жесткого диска на разделы

0 логический сектор - Master Boot Record (MBR):

- ◆ Код начального загрузчика
- ◆ Таблица разделов (4 записи)
- ◆ Маркер 0xAA55

Формат записи:

- Признак загрузки (0x80 или 0x00)
- Первый сектор раздела (C - H - S)
- Тип раздела (файловая система)
- Последний сектор раздела (C - H - S)
- Первый логический сектор раздела
- Размер



Extended boot record

- ◆ Для разбиения на большее число разделов в MBR указывается специальный тип раздела (extended)
- ◆ В начале раздела размещается сектор с таблицей разделов из двух строк
 - Первая описывает текущий раздел
 - Вторая является ссылкой на следующий extended раздел (если есть)

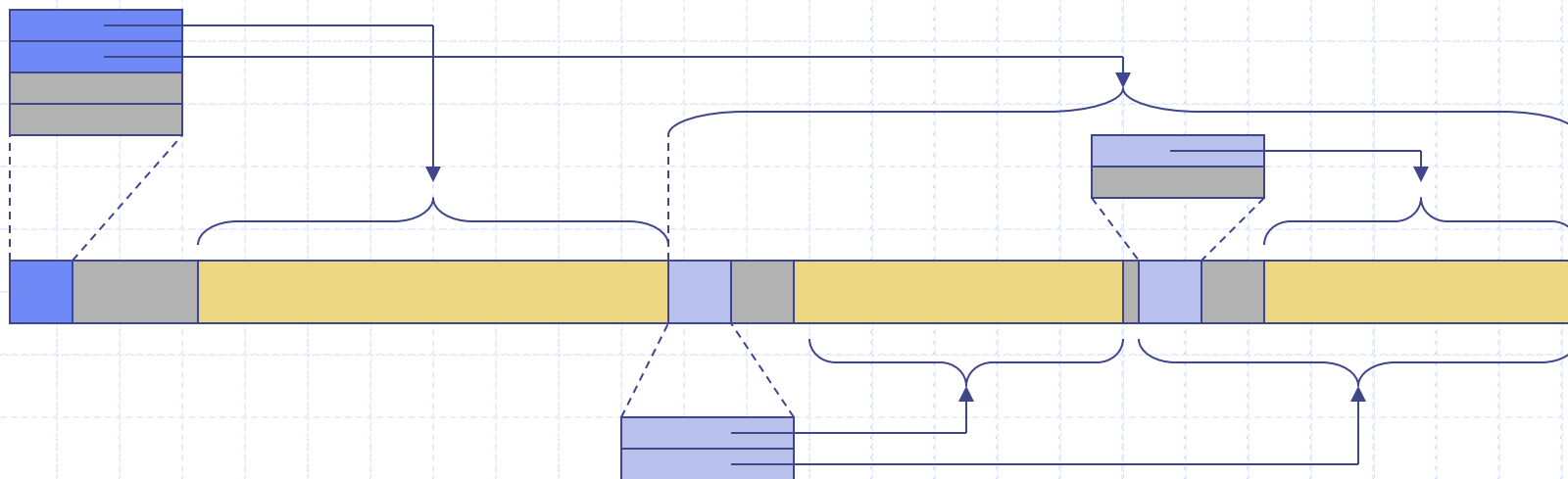


Таблица разделов GUID (GPT)

- ◆ Разработана Intel в конце 90-х как часть UEFI
- ◆ Использует адресацию LBA
- ◆ Использует 8 байт для хранения номеров секторов
- ◆ Содержит две копии таблицы разделов – в начале и конце диска

Сектор					
0	Защитная MBR				Основная GPT
1	Основной заголовок GPT				
2	Запись 1	Запись 2	Запись 3	Запись 4	
3	Записи 5-128				
34	Раздел 1				Резервная GPT
	Раздел 2				
	Остальные разделы				
-34					
-33	Запись 1	Запись 2	Запись 3	Запись 4	
-2	Записи 5-128				
-1	Резервный заголовок GPT				

Таблица разделов GUID (GPT)

Защитная MBR

- ◆ Используется для защиты раздела от систем, не поддерживающих GPT
- ◆ Содержит один раздел включающий первые 2 ТБ диска
- ◆ Тип раздела 0xEE

Сектор					
0	Защитная MBR				Основная GPT
1	Основной заголовок GPT				
2	Запись 1	Запись 2	Запись 3	Запись 4	
3	Записи 5-128				
34	Раздел 1				Резервная GPT
	Раздел 2				
	Остальные разделы				
-34					
-33	Запись 1	Запись 2	Запись 3	Запись 4	
-2	Записи 5-128				
-1	Резервный заголовок GPT				

Таблица разделов GUID (GPT)

Заголовок GPT

- ◆ Сигнатура "EFI PART"
- ◆ Версия
- ◆ Размер заголовка
- ◆ CRC32 заголовка
- ◆ Расположение заголовка
- ◆ Расположение копии
- ◆ Первый сектор, доступный для разделов
- ◆ Последний сектор, доступный для разделов
- ◆ GUID диска
- ◆ Начало таблицы разделов
- ◆ Число элементов таблицы разделов
- ◆ Размер записей таблицы разделов
- ◆ CRC32 таблицы разделов

Сектор					
0	Защитная MBR				Основная GPT
1	Основной заголовок GPT				
2	Запись 1	Запись 2	Запись 3	Запись 4	
3	Записи 5-128				
34	Раздел 1				
	Раздел 2				Резервная GPT
	Остальные разделы				
-34					
-33	Запись 1	Запись 2	Запись 3	Запись 4	
-2	Записи 5-128				
-1	Резервный заголовок GPT				

Таблица разделов GUID (GPT)

Записи таблицы разделов

- ◆ GUID типа раздела
- ◆ GUID раздела
- ◆ Первый сектор раздела
- ◆ Последний сектор раздела
- ◆ Атрибуты
 - Системный раздел
 - Загрузочный
 - Только чтение
 - Скрытый
 - Запрет автомонтирования
- ◆ Имя раздела

Сектор					
0	Защитная MBR				Основная GPT
1	Основной заголовок GPT				
2	Запись 1	Запись 2	Запись 3	Запись 4	
3	Записи 5-128				
34	Раздел 1				Резервная GPT
	Раздел 2				
	Остальные разделы				
-34					
-33	Запись 1	Запись 2	Запись 3	Запись 4	
-2	Записи 5-128				
-1	Резервный заголовок GPT				

Основные функции файловой системы

1. Идентификация файлов. Связывание имени файла с выделенным ему пространством внешней памяти.
2. Распределение внешней памяти между файлами.
3. Обеспечение надежности и отказоустойчивости.
4. Обеспечение защиты от несанкционированного доступа.
5. Обеспечение совместного доступа к файлам
6. Обеспечение высокой производительности.

Описание файла / metadata

- ◆ Имя файла
 - POSIX: 256 символов
- ◆ Тип устройства
- ◆ Тип файла
- ◆ Адресная информация
 - Начальный адрес
 - Размер (в байтах или блоках)
- ◆ Информация об управлении доступом
 - Владелец
 - Допустимые операции
- ◆ Информация об использовании
 - Дата создания
 - Дата последней модификации

Доступ к данным файла

◆ Последовательный доступ

- Подразумевает последовательное чтение/запись данных от начала до конца, с возможностью «перемотки» в начало.

◆ Произвольный (random) доступ

- Операции могут производиться начиная с любого места файла.

Структура файлов

- ◆ Неструктурированные файлы
 - Последовательность байтов
- ◆ Последовательности записей фиксированной длины
- ◆ Последовательность записей произвольной длины
 - Могут создаваться индексы для ускорения доступа по заданному значению ключа

Операции над файлами

- ◆ Создание файла
Устанавливается размещение и атрибуты файла
- ◆ Удаление файла
Освобождение занимаемого пространства
- ◆ Открытие файла
Проверка прав доступа и загрузка информации о расположении файла
- ◆ Закрытие файла
Освобождение системных ресурсов
- ◆ Позиционирование
Спецификация места следующих операций чтения/записи
- ◆ Чтение
- ◆ Запись

Последовательность действий над файлами

◆ Без сохранения состояния / Stateless

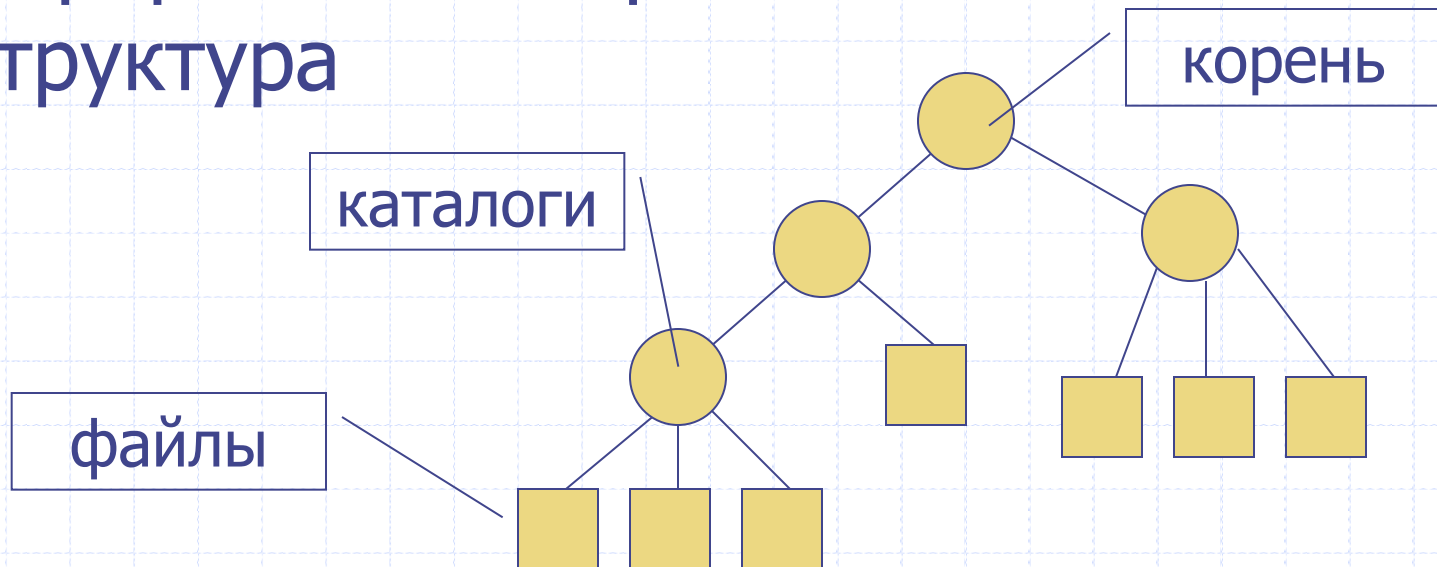
- Open, Read, Close, Open, Read, Close, Open, Write, Close

◆ С сохранением состояний

- Open, Read, Read, Write, Close

Логическая структура файловой системы

- ◆ Плоская
- ◆ Иерархическая с фиксированным числом уровней иерархии
- ◆ Иерархическая древовидная структура



Циклы в ФС

- ◆ Один файл может входить только в одну директорию

- ◆ Hard link

В разных директориях может быть записана информация об одном и том же файле.
Необходим счётчик ссылок.

- ◆ Soft link / Symbolic link

Создаётся специальный файл-ссылка. Оригинал может отсутствовать в момент создания.

Операции над директориями

- ◆ Создание
- ◆ Удаление
- ◆ Открытие
- ◆ Заккрытие
- ◆ Получение списка файлов
- ◆ Просмотр текущей записи
- ◆ Переименование файла
- ◆ Удаление файла

Реализация директорий

Файл с записями специальной структуры

◆ FAT

8	3	1	10	2	2	2	4
имя	расширение	атрибуты	резерв	время	дата	Первый блок	размер

◆ UNIX

2	14
i-node	имя

Поиск в директории

- ◆ Линейный поиск

- ◆ Хэш-таблицы

- ◆ Деревья

Семантические файловые системы

- ◆ Таксономия – древообразная структура классификаций набора объектов
- ◆ Фолксономия – совместная категоризация информации посредством произвольно выбираемых меток (тегов)
- ◆ WinFS
- ◆ Nepomuk
- ◆ DHTFS

Объединение нескольких ФС

Источники различных ФС:

- ◆ Разные носители
- ◆ Разделы на одном носителе

Существующие решения:

- ◆ Раздельный доступ к ФС
- ◆ Объединение в общую файловую систему (монтирование)

Защита файлов и ФС

- ◆ От разрушения

- ◆ От несанкционированного доступа

- Списки прав доступа
- Классификации пользователей

Права доступа к файлам в Unix

◆ Классы:

- Владелец
- Группа
- Остальные

◆ Права доступа:

- Чтение
- Запись
- Выполнение

Надёжность ФС

Обеспечение целостности ФС:

- ◆ Порядок выполнения операций
- ◆ Избыточность
- ◆ Журналирование
 - Как правило для системных структур
- ◆ Поддержка транзакций
- ◆ Проверка специальным утилитами

Работа FSCK в UNIX

1. Проверка блоков в файлах и списке свободных

1. Блока нигде нет:
2. Есть в списке свободных и в файле:
3. Входит в несколько файлов:

2. Сравнить счётчик ссылок на файл и ссылки

1. Счётчик больше чем есть ссылок:
2. Счётчик меньше

Управление плохими блоками

- ◆ На уровне контроллера диска
использование резервных блоков вместо плохих
- ◆ На уровне ОС
хранение списка повреждённых блоков,
например в виде файла