

ИНСТИТУТ
МАТЕМАТИКИ
МЕХАНИКИ
КОМПЬЮТЕРНЫХ
НАУК

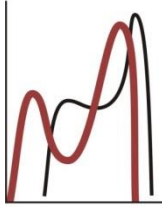
имени И.И. Воровича —

Архитектура компьютера и операционные системы

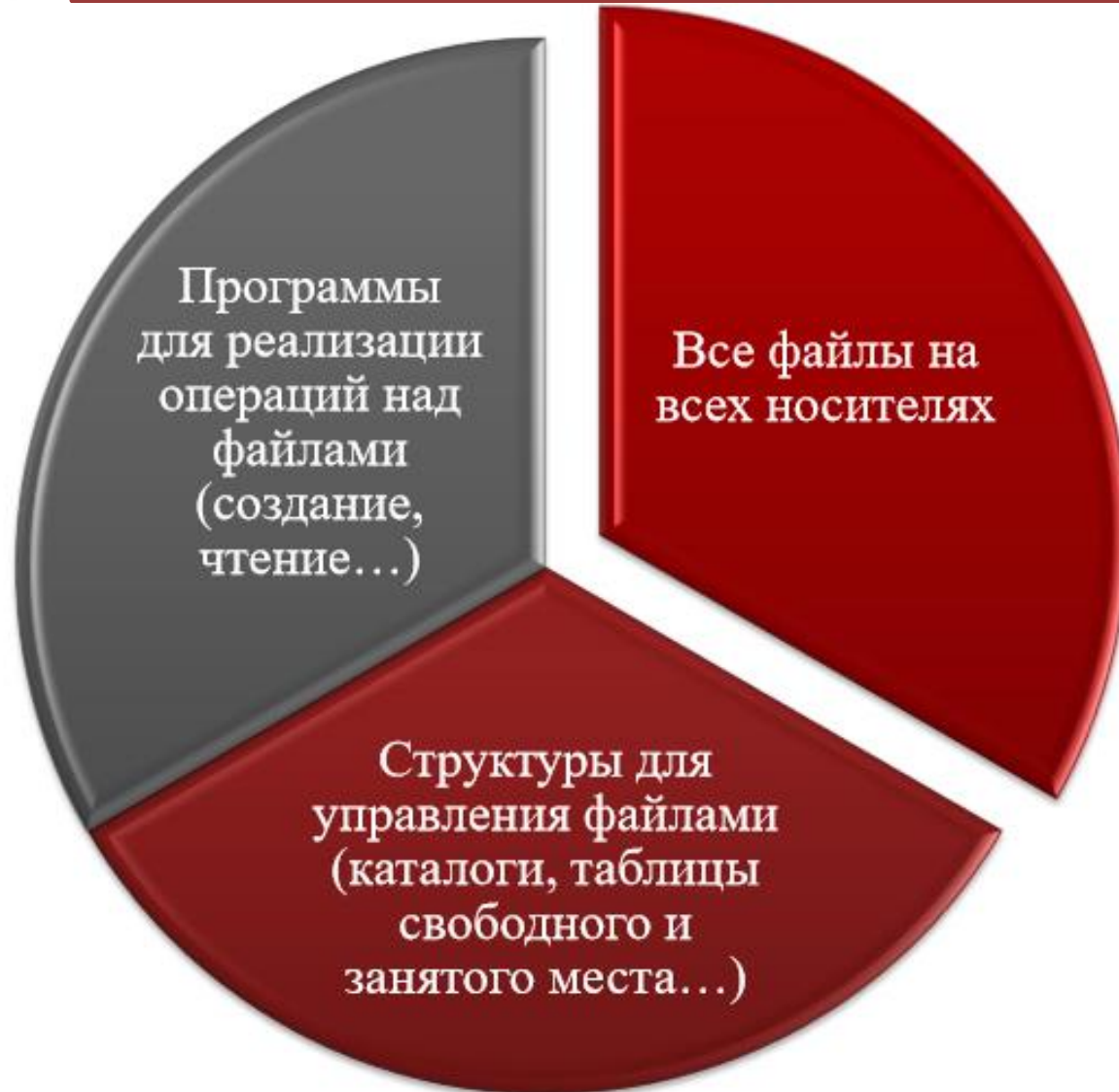
Лекция 19. Файловые системы.

Андреева Евгения Михайловна

доцент кафедры информатики и вычислительного эксперимента



Файловая система





Основная идея

- ОС делит внешнюю память на блоки фиксированного размера.
- Файл, обычно представляет собой неструктурированную последовательность записей
- Хранится файл в виде последовательности блоков (не обязательно смежных)
- Каждый блок хранит целое число записей.



Основные функции файловой системы

- Идентификация файлов.
 - Связывание имени файла с выделенным ему пространством внешней памяти.
- Распределение внешней памяти между файлами.
 - Для работы с конкретным файлом пользователю не требуется иметь информацию о местоположении этого файла на внешнем носителе информации.
- Обеспечение надежности и отказоустойчивости.
- Обеспечение защиты от несанкционированного доступа.
- Обеспечение совместного доступа к файлам.
 - Пользователю не нужно прилагать специальных усилий по обеспечению синхронизации доступа.
- Обеспечение высокой производительности.



Структура файловой системы

Система управления файлами

непрерывная последовательность блоков фиксированного размера

Система ввода-вывода

номер цилиндра, номер поверхности, номер блока на дорожке, число байт

Оборудование

Система управления файлами

- **Базисная подсистема**
 - алгоритмы выделения блоков диска и соответствующие структуры файла,
 - менеджер свободного пространства,
 - системные вызовы, работающие с дескриптором файла,
 - таблицы открытых файлов,
 - монтирование файловых систем,
 - реализация разделяемых файлов
- **Логическая подсистема**
 - поддержка иерархической древовидной структуры
 - системные вызовы, работающие с полным именем файла
 - защита файлов



Стандартный запрос

- Запрос на открытие или создание файла поступает от прикладной программы к логической подсистеме.
- Логическая подсистема, используя структуру директорий, проверяет права доступа и вызывает базовую подсистему для получения доступа к блокам файла.
- После этого файл считается открытым, он содержится в таблице открытых файлов
- Прикладная программа получает дескриптор этого файла.
- Дескриптор файла является ссылкой на файл в таблице открытых файлов и используется в запросах прикладной программы на чтение-запись из этого файла.
- Запись в таблице открытых файлов указывает через систему выделения блоков диска на блоки данного файла.
- Если к моменту открытия файл уже используется другим процессом, то есть содержится в таблице открытых файлов, то после проверки прав доступа к файлу может быть организован совместный доступ.
- При этом новому процессу также возвращается дескриптор



Общие сведения о файлах

- **Имена файлов.** Правила именования файлов зависят от ОС. Многие ОС поддерживают имена из двух частей: имя, расширение.
- **Типы файлов:**
 - Регулярные (обычные) файлы - набор блоков на устройстве внешней памяти, на котором поддерживается файловая система. Могут содержать как текстовую информацию (.c, .pas, .txt), так и произвольную двоичную (.obj, .zip)
 - Директории - системные файлы, поддерживающие структуру файловой системы.
- **Атрибуты файлов:**
 - основная информация (имя, тип файла), адресная информация (устройство, начальный адрес, размер), информация об управлении доступом (владелец, допустимые операции) и информация об использовании (даты создания, последнего чтения, модификации и др.).



Организация файлов и доступ к ним

- Последовательный файл
- Файл прямого доступа
- Файл в виде последовательности записей переменной длины
- Индексированный файл



Последовательный файл

- Так как файл является последовательностью записей (однобайтовых), то файл представляет собой неструктурированную последовательность байтов.
- Обработка предполагает последовательное чтение записей от начала файла
- Операции над файлами:
 - Чтение с текущей позиции
 - Запись в текущую позицию
 - Возврат на начало файла
- Доступ к файлу последовательный





Файл прямого доступа

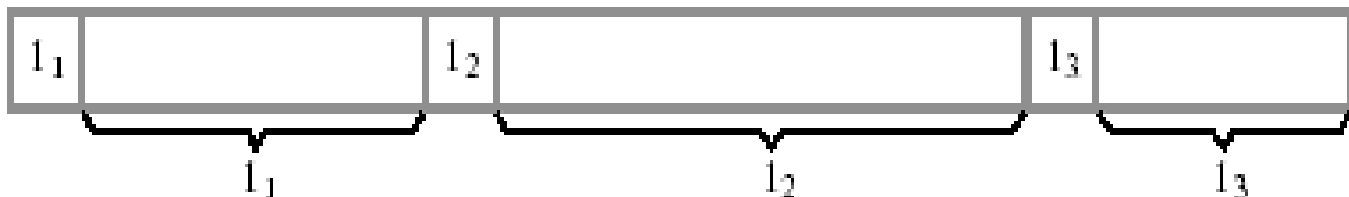
- Файл, байты которого могут быть считаны в произвольном порядке, называется файлом прямого доступа.
- Операции над файлами:
 - Поиск нужного места внутри файла (seek)
 - Чтение с текущей позиции
 - Запись в текущую позицию
- Доступ к файлу произвольный





Файл в виде последовательности записей переменной длины

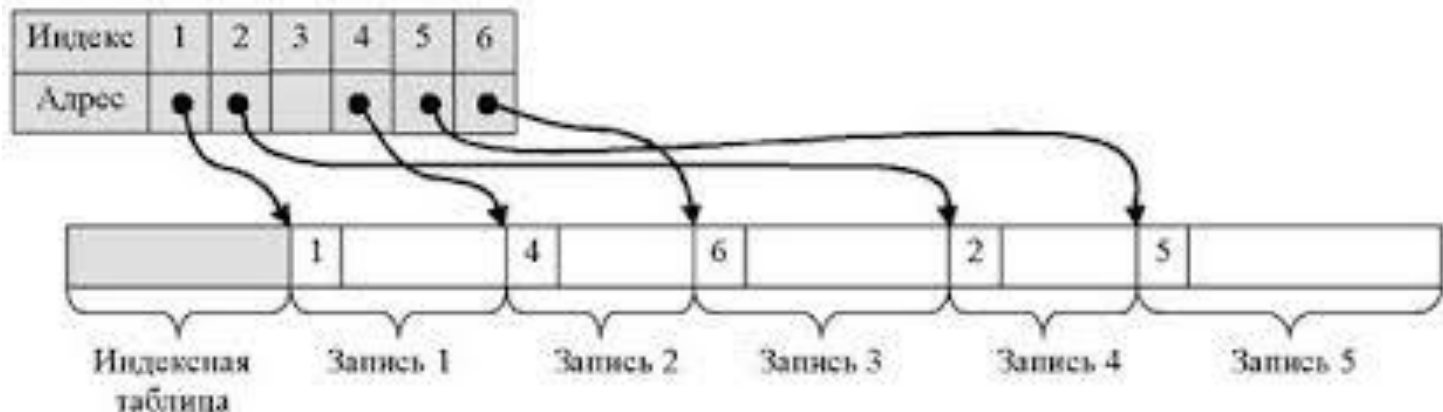
- Запись содержит ключевое поле в фиксированной позиции
- Базовая операция - считать запись с каким-либо значением ключа.
- Записи могут располагаться в файле последовательно или в более сложном порядке.
- Метод доступа по значению ключевого поля к записям последовательного файла называется индексно-последовательным.





Индексированный файл

- Строится индекс файла, который состоит из списка элементов, каждый из которых содержит идентификатор записи, за которым следует указание о местоположении данной записи
- Для поиска записи вначале происходит обращение к индексу, где находится указатель на нужную запись.
- Метод доступа - доступ с использованием индекса.





Операции над файлами

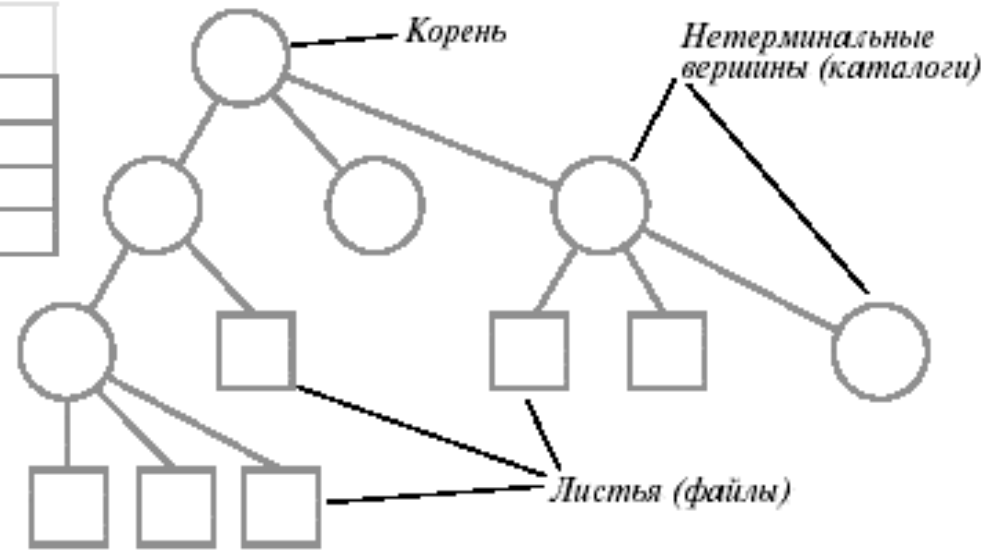
- Создание файла
- Удаление файла
- Открытие файла
- Заккрытие файла
- Позиционирование
- Чтение данных из файла
- Запись данных в файл с текущей позиции



Древовидная структура файловой системы

- Все современные файловые системы поддерживают многоуровневое именование файлов за счет наличия во внешней памяти дополнительных файлов со специальной структурой - каталогов (или директорий)

Имя файла (каталога)	Тип файла (обычный или каталог)	
Anti	К	атрибуты
Games	К	атрибуты
Autoexec.bat	О	атрибуты
mouse.com	О	атрибуты





Операции над директориями

- Создание директории
- Удаление директории
- Открытие директории для последующего чтения
- Заккрытие директории
- Поиск
- Получение списка файлов
- Переименование
- Создание файла
- Удаление файла



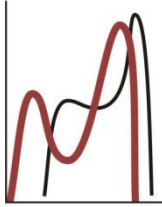
Контроль доступа к файлам

- Списки прав доступа
- Классификации пользователей (9 бит - rwxrwxrwx).
 - Владелец (Owner).
 - Группа (Group).
 - Остальные (Others).



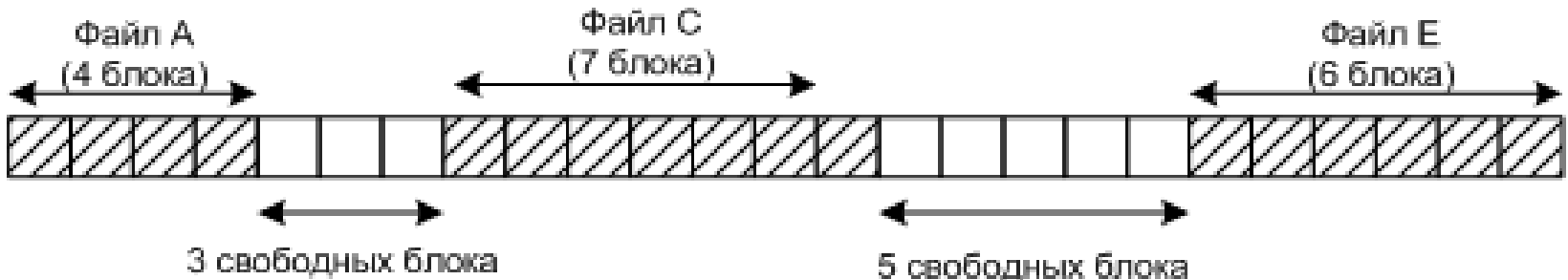
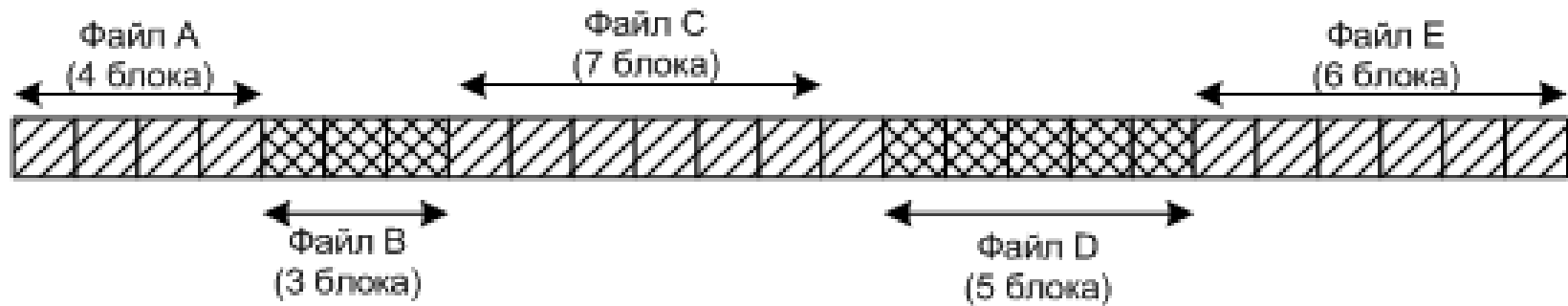
Методы выделения дискового пространства

- Выделение непрерывной последовательностью блоков
- Связный список
- Таблица отображения файлов
- Индексные узлы



Выделение непрерывной последовательностью блоков

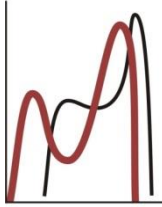
- Файл хранится как непрерывная последовательность блоков диска
- Характеризуется адресом и длиной





Преимущества и недостатки

- Легко реализуется
 - выяснение местонахождения файла сводится к вопросу, где находится первый блок.
- Высокая производительность
 - целый файл может быть считан за одну дисковую операцию.
- Внешняя фрагментация
- Проблемы при увеличении размера файла



СВЯЗНЫЙ СПИСОК

- В начале каждого блока содержится указатель на следующий блок.
- Адрес файла может быть задан одним числом - номером первого блока.





Недостатки

- Сложность реализации доступа к произвольно заданному месту файла
- Количество данных файла, содержащихся в одном блоке, не равно степени двойки, а многие программы читают данные блоками, размер которых равен степени двойки.
- Низкая надежность

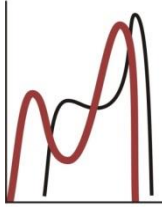
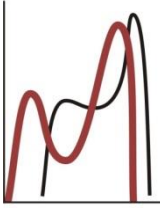


Таблица отображения файлов

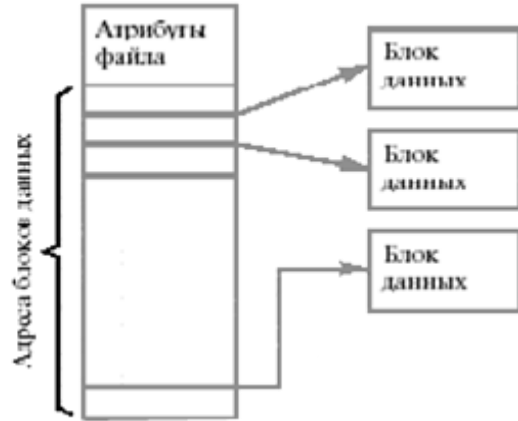
FAT - file allocation table

Номера блоков диска		
1		
2	10	
3	11	Начало файла F ₂
4		
5	EOF	
6	2	Начало файла F ₁
7	EOF	
8		
9		
10	7	
11	5	

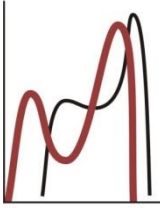
- Указатели хранятся не в дисковых блоках, а в индексной таблице в памяти
- Запись в директории содержит только ссылку на первый блок
- По таблице отображения можно судить о физическом соседстве блоков, располагающихся на диске
- Минусом данной схемы является необходимость хранения в памяти этой довольно большой таблицы.



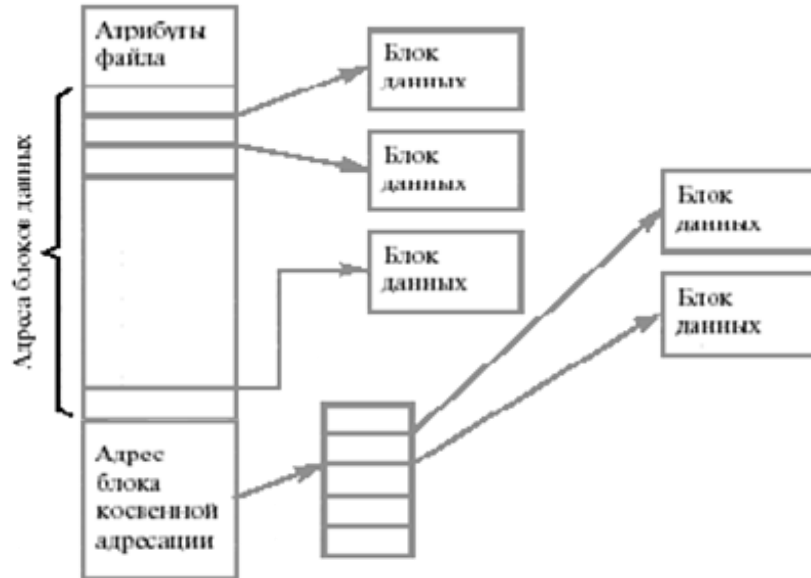
Индексные узлы (i-node)



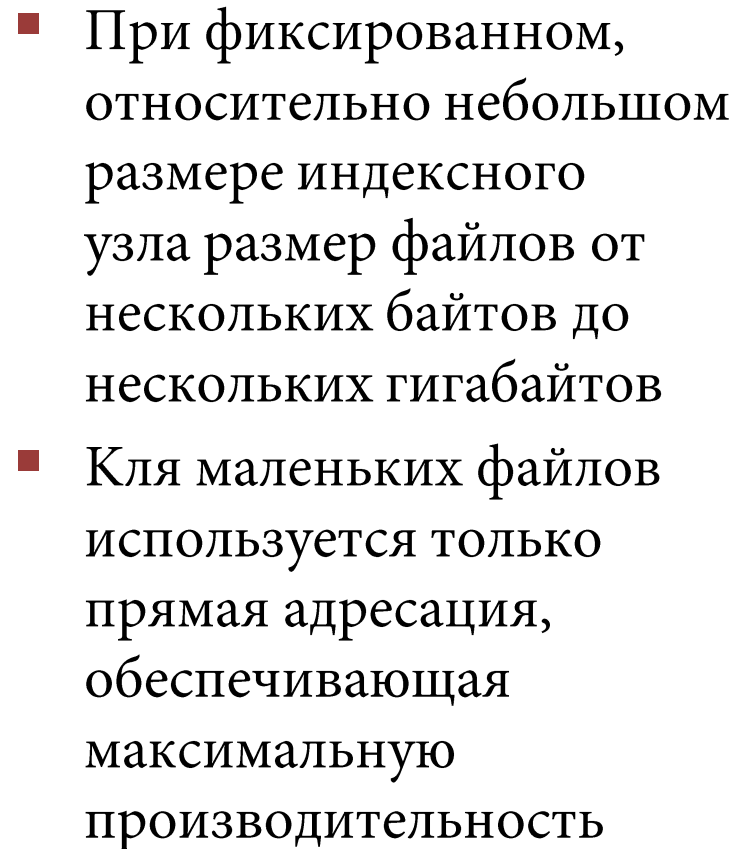
- При фиксированном, относительно небольшом размере индексного узла размер файлов от нескольких байтов до нескольких гигабайтов
- Для маленьких файлов используется только прямая адресация, обеспечивающая максимальную производительность

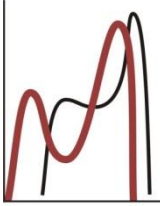


Индексные узлы (i-node)

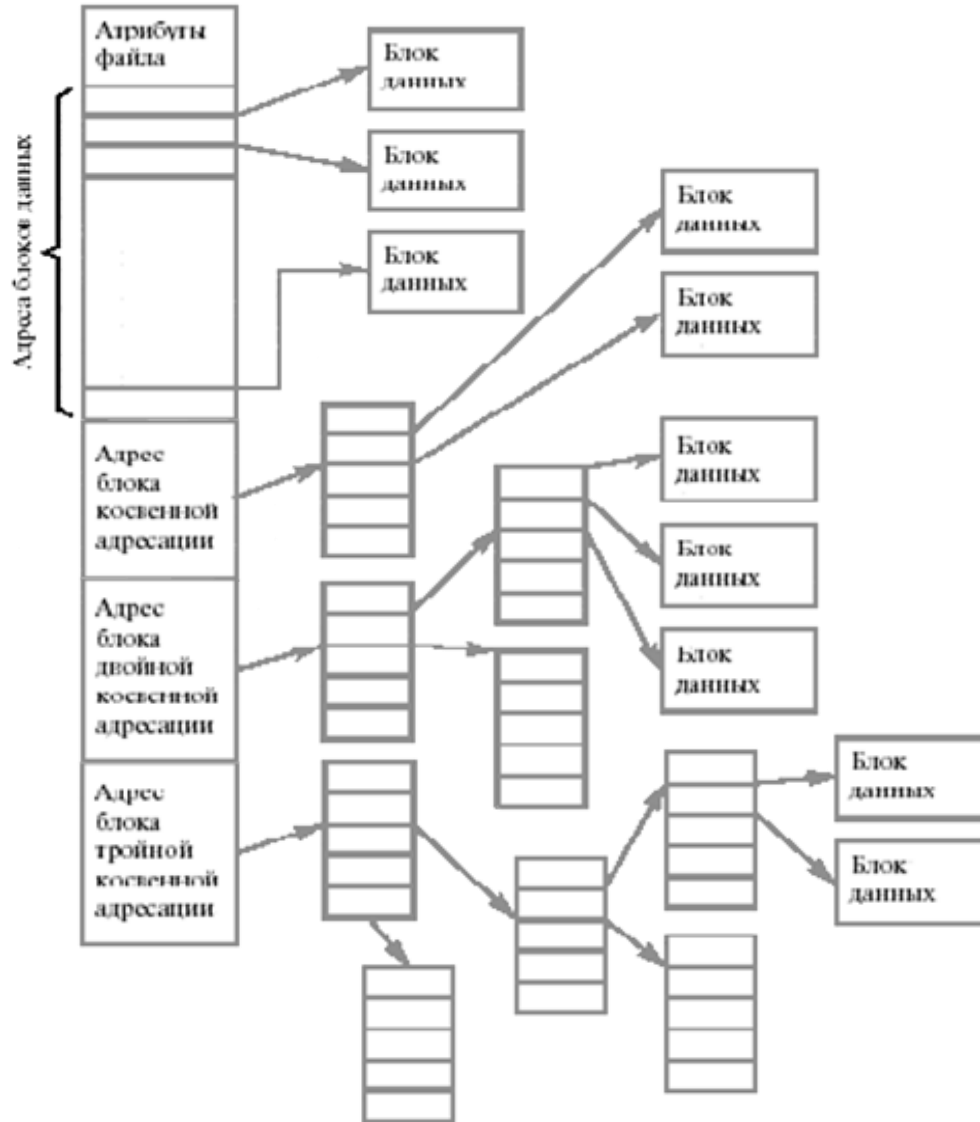


- При фиксированном, относительно небольшом размере индексного узла размер файлов от нескольких байтов до нескольких гигабайтов
- Для маленьких файлов используется только прямая адресация, обеспечивающая максимальную производительность

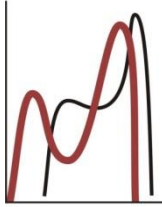




Индексные узлы (i-node)



- При фиксированном, относительно небольшом размере индексного узла размер файлов от нескольких байтов до нескольких гигабайтов
- Для маленьких файлов используется только прямая адресация, обеспечивающая максимальную производительность



Реализация директорий

- Директория или каталог - это файл, имеющий вид таблицы и хранящий список входящих в него файлов или каталогов.
- Основная задача файлов-директорий - поддержка иерархической древовидной структуры файловой системы.
- Запись в директории имеет определенный для данной ОС формат, поэтому блоки данных файла-директории заполняются при помощи специальных системных вызовов (например, создание файла).

Имя файла (каталога)	Тип файла (обычный или каталог)	
Anti	K	атрибуты
Games	K	атрибуты
Autoexec.bat	O	атрибуты
mouse.com	O	атрибуты

Структуры,
содержащие
адреса
блоков
файлов

Примеры реализации директорий

■ Директории в ОС MS-DOS

Номер первого блока используется в качестве индекса в таблице *FAT*

Разряды	8	3	1	10	2	2	2	4
	Имя файла	Расширение	Атрибуты (обычный файл или директория)	Резервное поле	Время	Дата	Номер первого блока	Размер

■ Директории в ОС Unix

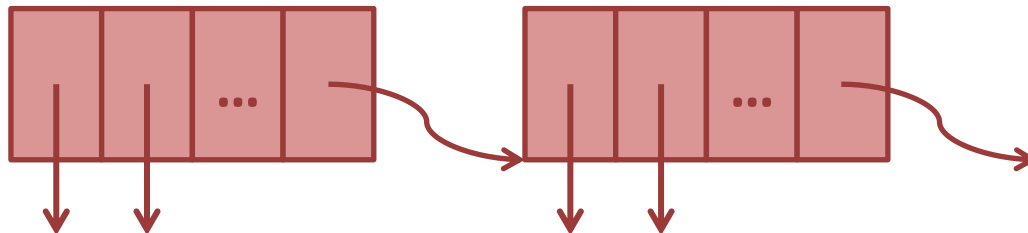
Каждая запись содержит имя файла и номер его индексного узла

Байты	2	14
	Номер индексного узла	Имя файла



Управление свободным дисковым пространством

- Учет при помощи организации битового вектора
 - Каждый блок представлен одним битом, принимающим значение 0 или 1, в зависимости от того, занят он или свободен.
 - Например, 00111100111100011000001 ...
- Учет при помощи организации связного списка
 - В список связываются все свободные блоки, указатель на первый свободный блок размещается в специально отведенном месте диска
 - Организуется хранение адресов n свободных блоков в первом свободном блоке. Первые $n-1$ этих блоков действительно используются. Последний блок содержит адреса других n блоков



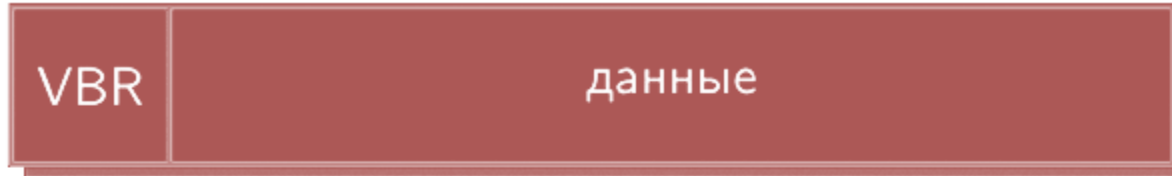


Разделы жесткого диска

- **Раздел** — часть долговременной памяти жёсткого диска или флеш-накопителя, выделенная для удобства работы, и состоящая из смежных блоков. На одном устройстве хранения может быть несколько разделов.
- Преимущества использования нескольких разделов
 - на одном жёстком диске можно установить несколько операционных систем
 - можно отделить информацию пользователя от файлов операционной системы
 - на одном физическом жёстком диске можно хранить информацию в разных файловых системах
 - манипуляции с одной файловой системой не сказываются на других файловых системах



Организация разделов физического диска



первичный раздел

расширенный раздел



Дескрипторы разделов

- Главная загрузочная запись: (Master Boot Record, MBR) — находится в первом физическом секторе (секторах), содержит:
 - код загрузки (NSB :Non -System Bootstrap);
 - таблицу разделов (Partition table) — до 4;
 - сигнатуру 55AAh . . .
- Загрузочная запись тома: (Volume Boot Record, VBR) — зависит от типа файловой системы на логическом разделе диска и содержит код, выполняющий нахождение и загрузку операционной системы на данном типе файловой системы.
- Расширенная загрузочная запись: (Extended Boot Record, EBR) — находится в начале логического раздела, структура, как у MBR, но запись для II раздела указывает на следующую EBR.

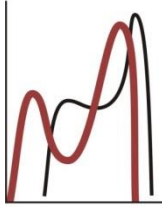
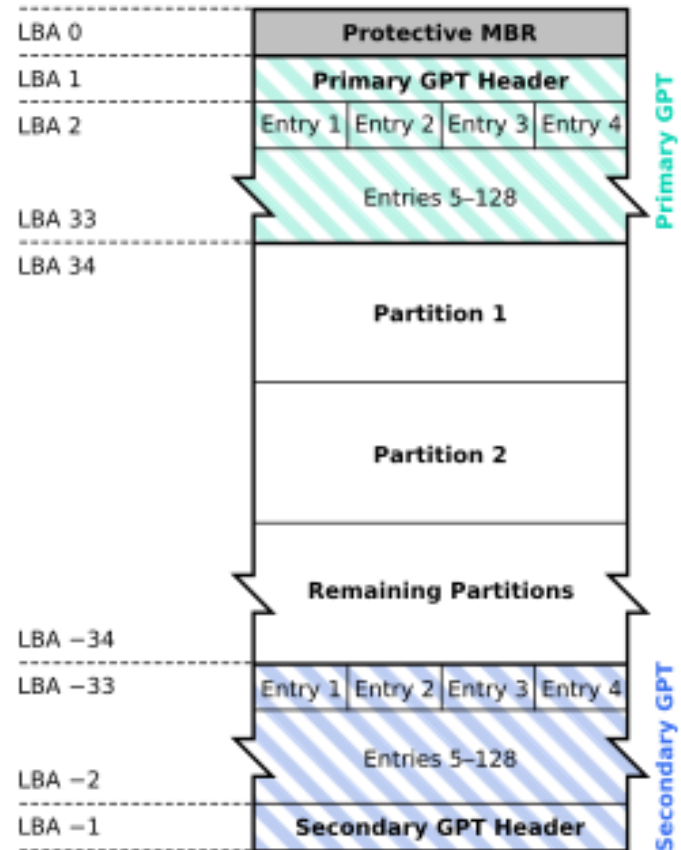
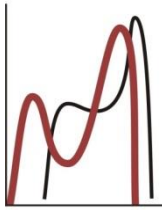


Таблица разделов GPT

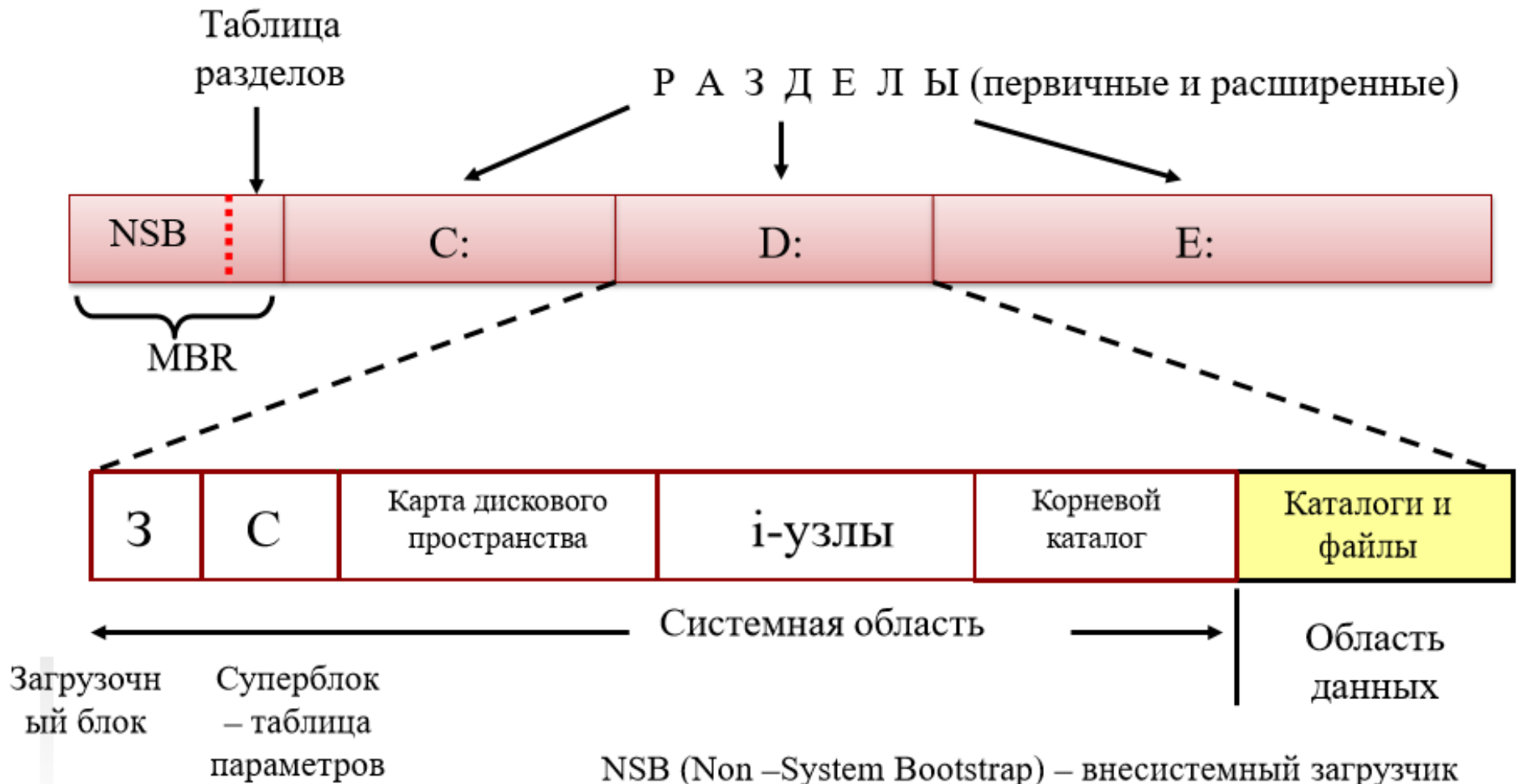
- Стандарт формата размещения таблиц разделов (часть Unified Extensible Firmware Interface — UEFI);
- Windows до 128 разделов, GNU/Linux до 256
- Ограничение на размер диска: 8 Збайт - 512 байт (у MBR: 2 Тбайт - 512 байт);
- Заголовок GPT и таблица разделов хранятся в начале и конце диска;
- «Защитный MBR» в начале диска;
- GPT использует современную систему адресации логических блоков (LBA) вместо применявшейся в MBR адресации «Цилиндр — Головка — Сектор» (CHS).

GUID Partition Table Scheme





Структура файловой системы на диске





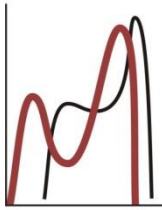
Суперблок

- тип файловой системы;
- размер файловой системы в блоках;
- размер массива индексных узлов
 - определяется администратором при установке системы.
Максимальное число файлов, которые могут быть созданы в файловой системе, определяется числом доступных индексных узлов;
- размер логического блока
 - может задаваться при форматировании файловой системы.



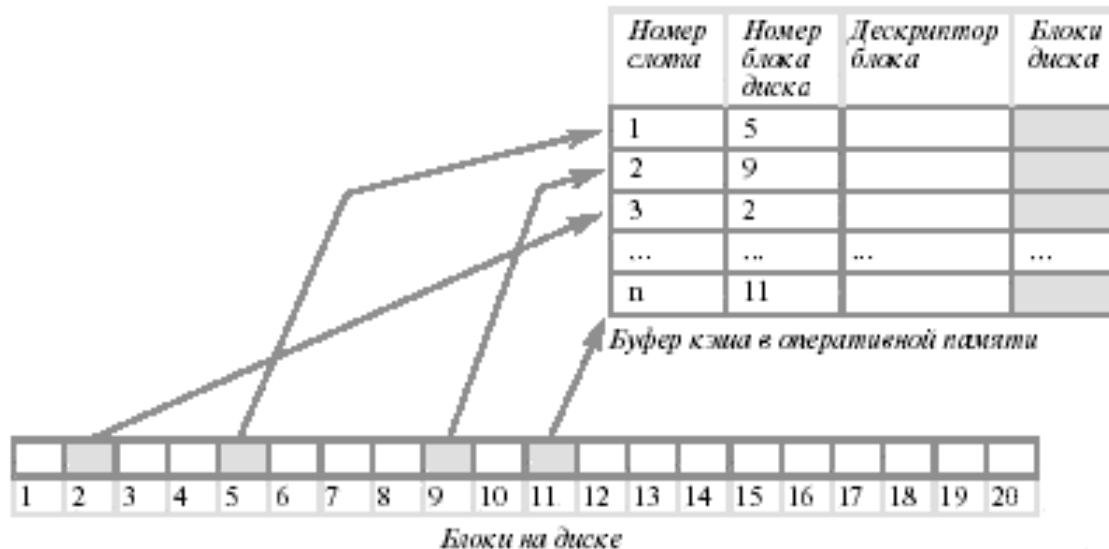
Надежность файловой системы

- Целостность файловой системы
- Проверка целостности файловой системы при помощи утилит
 - fsck, chkdsk, scandisk и др.
- Порядок выполнения операций
- Журнализация
- Управление "плохими" блоками

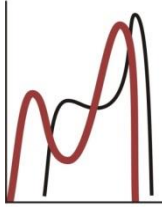


Производительность файловой системы

- Кэширование
 - емкость буфера кэша ограничена
 - несвоевременная синхронизация



- Оптимальное размещение информации на диске



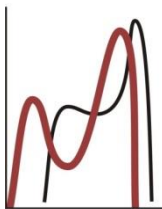
Современные архитектуры файловых систем





Часто используемые ФС

- **ext4**: журналируемая файловая система, разработана специально для ядра Linux. Является потомком более старых версий ext3, ext2 и ext.
- **NTFS**: используется в качестве основной для Windows NT. Имеет поддержку хранения дескрипторов безопасности.
- **FAT**: использовалась в MS-DOS, позже — в Windows 3.11, Windows 95 и т. д. Используется в портативных устройствах и съёмных носителях. Не имеет поддержки разграничения прав доступа. Последней версией является FAT32, (размер файла ≤ 4 ГБ).
- **HFS+**: файловая система, разработанная Apple Inc. для замены HFS. Является улучшенной версией HFS, с поддержкой файлов большого размера
- **ISO 9660** (или CDFS): используется для хранения файлов на носителях CD-ROM, DVD-ROM и т. д. (размер файла ≤ 4 ГБ).
- **UDF**: разработана для носителей CD-RW, DVD и т. д. В настоящее время имеет поддержку жёстких дисков и носителей на Flash-памяти.



Основные характеристики файловых систем

	Разрядность указателя	Число кластеров	Максимальн ый объем кластера	Максимальн ый размер раздела	Имя файла
FAT12	12	4096	4 Кбайт	16 Мбайт	8.3
FAT16	16	65536	64 Кбайт	4 Гбайт	8.3
FAT 32	32	4 Гбайт	32 Кбайт	2^{32} по 32 Кбайт	255.3
NTFS	64	2^{64}	4 Кбайт	2^{64} по 4 Кбайт	255.3



Основные возможности NTFS

- Размеры блоков: от 512 байт до 64К.
- Максимальный размер диска: 2^{64} блока.
- Максимальный размер файла: 2^{64} байт.
- Альтернативные потоки данных.
- Квоты.
- Разреженные файлы.
- Монтирование.
- Жесткие ссылки.
- Сжатие файлов.
- Шифрование данных.
- Журналирование.



Домашнее задание

- Подготовка к тестированию по лекциям.
- Читать книгу Таненбаум Э., Бос Х. Современные операционные системы, стр.301-380.