

1. Понятие ОС. Задачи ОС. ОС как виртуальная машина. ОС как система управления ресурсами.

Операционная система (ОС) — это комплекс системных программ, управляющий работой аппаратных и программных средств компьютера и обеспечивающий взаимодействие с пользователем.

Основные задачи ОС:

- Управление ресурсами (процессор, память, устройства ввода/вывода).
- Предоставление интерфейса для пользователя и приложений.
- Обеспечение надежности, безопасности и защиты данных.
- Поддержка многозадачности и многопользовательской работы.

ОС как виртуальная машина:

ОС предоставляет абстрактную машину, скрывающую сложность реального оборудования от пользователя и приложений. Это позволяет создавать единое пространство для запуска программ независимо от конкретной аппаратной платформы.

ОС как система управления ресурсами:

ОС распределяет процессорное время, оперативную память, дисковое пространство и другие ресурсы между выполняющимися процессами, обеспечивая их эффективное использование и предотвращение конфликтов.

2. История и эволюция ОС

Развитие ОС можно разделить на несколько этапов:

1-е поколение (конец 1940-х — начало 1950-х годов)

- **Аппаратная база:** электронно-вакуумные лампы.
- **Особенности программирования:**
- Программы загружались вручную.
- Не существовало понятия ОС.
- Каждая программа была уникальной и требовала ручного управления оборудованием.
- **Режим работы:** однопрограммный, без разделения времени.
- **Примеры компьютеров:** ENIAC, EDSAC.

2-е поколение (середина 1950-х — начало 1960-х годов)

- **Аппаратная база:** транзисторы.
- **Начало появления первых управляющих программ.**
- **Появились:**
- Библиотеки подпрограмм.
- Программы-мониторы, которые обеспечивали автоматическую последовательную обработку заданий.
- **Режим работы:** пакетная обработка заданий.
- **Цель:** уменьшить время между выполнением программ.
- **Примеры ОС:** IBM FORTRAN Monitor System.

3-е поколение (1960-е — 1970-е годы)

- **Аппаратная база:** интегральные схемы.
- **Основные достижения:**
- Внедрение мультипрограммирования (одновременное хранение нескольких программ в памяти).
- Появление систем разделения времени (time-sharing), позволяющих нескольким пользователям работать с одной машиной одновременно.
- Развитие файловых систем и защиты данных.
- **Примеры ОС:**
- **IBM OS/360** — одна из первых универсальных операционных систем, созданная для широкого спектра оборудования.
- **Multics** — экспериментальная система, оказавшая влияние на развитие UNIX.
- **Изменения в пользовательском интерфейсе:** командный интерфейс стал более развитым.

4-е поколение (с 1980-х годов по настоящее время)

- **Аппаратная база:** микропроцессоры, большие интегральные схемы.
- **Важнейшие изменения:**
- Распространение персональных компьютеров (ПК).
- Появление графического интерфейса пользователя (GUI) — например, Apple Macintosh, Microsoft Windows.
- Развитие сетевых технологий и поддержка многозадачности и многопользовательской работы.
- Интеграция с интернетом и облачными технологиями.
- **Примеры ОС:**
- **MS-DOS** — популярная однозадачная ОС для ПК.
- **Windows** — семейство графических ОС от Microsoft.
- **UNIX/Linux/macOS** — мощные многозадачные, многопользовательские системы.
- **Android/iOS** — мобильные ОС.
- **Современные тенденции:**
- Поддержка виртуализации.
- Облачные технологии.
- Высокий уровень безопасности и автоматизации.
- Искусственный интеллект в управлении системами.

Современные ОС:

- Поддержка мобильных устройств, облачных технологий, виртуализации.
- Высокий уровень безопасности, мультимедийных возможностей, автоматизации.

3. Классификация ОС по особенностям управления ресурсами. Режимы многозадачности и многопользовательности.

От эффективности алгоритмов управления локальными ресурсами компьютера во многом зависит эффективность всей сетевой ОС в целом. Поэтому, характеризуя сетевую ОС, часто приводят важнейшие особенности реализации функций ОС по управлению

процессорами, памятью, внешними устройствами автономного компьютера. Так, например, в зависимости от особенностей использованного алгоритма управления процессором, операционные системы делят на многозадачные и однозадачные, многопользовательские и однопользовательские, на системы, поддерживающие многопотоковую обработку и не поддерживающие ее, на многопроцессорные и однопроцессорные системы.

Поддержка многозадачности.

По числу одновременно выполняемых задач операционные системы могут быть разделены на два класса:

однозадачные (например, MS-DOS, MSX) и

многозадачные (OS EC, OS/2, UNIX, Windows 95).

Однозадачные ОС в основном выполняют функцию предоставления пользователю виртуальной машины, делая более простым и удобным процесс взаимодействия пользователя с компьютером. Однозадачные ОС включают средства управления периферийными устройствами, средства управления файлами, средства общения с пользователем.

Многозадачные ОС, кроме вышеперечисленных функций, управляют разделением совместно используемых ресурсов, таких как процессор, оперативная память, файлы и внешние устройства.

Поддержка многопользовательского режима.

По числу одновременно работающих пользователей ОС делятся на:

однопользовательские (MS-DOS, Windows 3.x, ранние версии OS/2);

многопользовательские (UNIX, Windows NT).

Главным отличием многопользовательских систем от однопользовательских является наличие средств защиты информации каждого пользователя от несанкционированного доступа других пользователей. Следует заметить, что не всякая многозадачная система является многопользовательской, и не всякая однопользовательская ОС является однозадачной.

4. Классификация ОС по способам распределения процессорного времени. вытесняющие и не вытесняющие алгоритмы.

Важнейшим разделяемым ресурсом является процессорное время. Способ распределения процессорного времени между несколькими одновременно существующими в системе процессами (или нитями) во многом определяет специфику ОС. Среди множества существующих вариантов реализации многозадачности можно выделить две группы алгоритмов:

невывтесняющая многозадачность (NetWare, Windows 3.x);

вывтесняющая многозадачность (Windows NT, OS/2, UNIX).

Основным различием между вытесняющим и невытесняющим вариантами многозадачности является степень централизации механизма планирования процессов. В первом случае механизм планирования процессов целиком сосредоточен в операционной системе, а во втором - распределен между системой и прикладными программами. При невытесняющей многозадачности активный процесс выполняется до тех пор, пока он сам, по собственной инициативе, не отдаст управление операционной системе для того, чтобы та выбрала из очереди другой готовый к выполнению процесс. При вытесняющей многозадачности решение о переключении процессора с одного процесса на другой принимается операционной системой, а не самим активным процессом.

5. Классификация ОС по архитектуре вычислительных систем. Сетевые, многопроцессорные и кластерные ОС

- Сетевые ОС :
Предоставляют средства для подключения к сети, удаленного доступа и обмена данными.
- Многопроцессорные ОС :
Управляют несколькими процессорами, обеспечивая параллелизм и балансировку нагрузки.
- Кластерные ОС :
Представляют собой объединение нескольких компьютеров в единый вычислительный ресурс.

Сетевая ОС имеет в своем составе средства передачи сообщений между компьютерами по линиям связи, которые совершенно не нужны в автономной ОС.

На основе этих сообщений сетевая ОС поддерживает разделение ресурсов компьютера между удаленными пользователями, подключенными к сети. Для поддержания функций передачи сообщений сетевые ОС содержат специальные программные компоненты, реализующие популярные коммуникационные протоколы, такие как TCP/IP, IPX, Ethernet и другие.

Многопроцессорные системы требуют от операционной системы особой организации, с помощью которой сама ОС, а также поддерживаемые этой ОС приложения могли бы выполняться параллельно отдельными процессорами системы.

Параллельная работа отдельных частей ОС создает дополнительные проблемы для разработчиков ОС, так как в этом случае гораздо сложнее обеспечить согласованный доступ отдельных процессов к общим системным таблицам, исключить нежелательные последствия асинхронного выполнения работ.

Операционная система кластеров

Другие требования предъявляются к операционным системам *кластеров*.

Кластер — *слабо связанная совокупность нескольких вычислительных систем, работающих совместно для выполнения общих приложений, и представляющихся пользователю единой системой.*

Наряду со специальной аппаратурой для функционирования кластерных систем необходима и программная поддержка со стороны ОС, которая сводится в основном к синхронизации доступа к разделяемым ресурсам, обнаружению отказов и динамической реконфигурации системы.

Одной из первых разработок в области кластерных технологий были решения компании Digital Equipment Corporation на базе компьютеров VAX. Недавно этой компанией заключено соглашение с корпорацией Microsoft о разработке кластерной технологии, использующей Windows NT. Несколько компаний предлагают кластеры на основе Unix-машин.

6. Основные типы многозадачных ОС: пакетной обработки, разделения времени и реального времени.

- о системы пакетной обработки (batch processing operating systems) – целью является выполнение максимального количества вычислительных задач за единицу времени; при этом из нескольких задач формируется пакет, который обрабатывается системой;
- о системы разделения времени (time-sharing operating systems) – целью является возможность одновременного использования одного компьютера несколькими пользователями; реализуется посредством поочередного предоставления каждому пользователю интервала процессорного времени;
- о системы реального времени (real-time operating systems) – целью является выполнение каждой задачи за строго определённый для данной задачи интервал времени.

7. Способы построения ядра ОС: монолитное ядро, микро ядро.

Существует два основных вида ядер – монолитные ядра (monolithic kernel) и микроядра (microkernel).

В **монолитном** ядре реализуются все основные функции операционной системы, и оно является, по сути, единой программой, представляющей собой совокупность процедур [6].

В **микроядре** остается лишь минимум функций, который должен быть реализован в привилегированном режиме: планирование потоков, обработка прерываний, межпроцессное взаимодействие. Остальные функции операционной системы по управлению приложениями, памятью, безопасностью и пр. реализуются в виде отдельных модулей в пользовательском режиме.

Ядра, которые занимают промежуточное положение между монолитными и микроядрами, называют **гибридными** (hybrid kernel).

- **Монолитное ядро :**

Все компоненты ОС работают в режиме ядра. Быстрее, но менее устойчиво к сбоям. Примеры: UNIX, Linux.

- **Микроядро :**

Только базовые функции (планирование, межпроцессное взаимодействие)

находятся в ядре. Остальное — в пользовательском режиме. Более устойчиво, но медленнее. Пример: MINIX, QNX.

8. Структура программного обеспечения ЭВМ. Компоненты системного программного обеспечения.

- **Структура ПО :**
- Прикладное ПО — программы для решения конкретных задач.
- Инструментальное ПО — компиляторы, отладчики.
- Системное ПО — ОС, драйверы, утилиты.
- **Компоненты системного ПО :**
- Ядро ОС.
- Драйверы устройств.
- Системные утилиты.
- Интерфейс командной строки и GUI.

10. Основные функции ОС.

- Управление процессором (планирование задач).
- Управление памятью (виртуализация, страничная/сегментная организация).
- Управление вводом-выводом.
- Файловая система.
- Предоставление пользовательского интерфейса.
- Обеспечение безопасности.
- Сетевая поддержка.

11. Понятие вычислительного ресурса.

Понятие ресурса строго не определено. Будем считать, что всякий потребляемый объект (независимо от формы его существования), обладающий некоторой практической ценностью для потребителя, является *ресурсом* [12].

Ресурсы различаются по запасу выделяемых единиц ресурса и бывают в этом смысле *исчерпаемыми* и *неисчерпаемыми*.

К *исчерпаемым* ресурсам относится, например, центральный процессор.

В качестве *неисчерпаемого* ресурса можно представить, например, память, выделяемую программе, если рассматривать ее как совокупность всех имеющихся в компьютере запоминающих устройств. В то же время, запоминающее устройство, состоящее только из

оперативной памяти с единственным трактом записи/считывания, представляет собой исчерпаемый ресурс.

Исчерпаемость ресурса, как правило, приводит к конфликтам среди потребителей этого ресурса. Для регулирования конфликтов ресурсы должны распределяться между потребителями по каким-то правилам, в наибольшей степени их удовлетворяющим.

12. Процессы. Порождение процессов. Идентификаторы процессов.

- **Процесс** — экземпляр выполняющейся программы.
- **Порождение процессов** :
Процесс порождает новый процесс через системный вызов `fork()` или аналогичный.
- **Идентификатор процесса (PID)** — уникальный номер, присваиваемый каждому процессу.

13. Взаимосвязи между родительскими и порожденными процессами.

- Процесс, создавший другой процесс, называется **родительским** .
- Созданный процесс — **дочерним** .
- Они могут совместно использовать данные или быть полностью изолированными, в зависимости от политики ОС.

Источник: Лекция 2 – *Управление процессами* .

14. Контекст процесса.

- **Контекст процесса** — состояние процессора и среды, необходимое для возобновления выполнения процесса после прерывания.
- Включает: регистры, программный счетчик, указатели стека, таблицы страниц, открытые файлы, состояние ввода-вывода.

Источник: Лекция 2 – *Управление процессами* .

15. Основные режимы процесса. Состояние процессов.

- **Режимы процесса** :
 - Пользовательский режим (user mode).
 - Режим ядра (kernel mode).
- **Состояния процессов** :
 - **Выполнение** — процесс выполняется на процессоре.
 - **Готовность** — процесс ожидает своей очереди.
 - **Ожидание** — процесс ждет завершения операции ввода-вывода или события.

Источник: Лекция 2 – *Управление процессами* .

16. Архитектура ядра ОС. Основные подсистемы ядра.

- **Ядро ОС** — центральный модуль, управляющий всеми ресурсами.
- **Основные подсистемы ядра** :
- Управление процессами и потоками.
- Управление памятью.
- Управление вводом-выводом.
- Файловая система.
- Сеть.
- Безопасность.

Источник: Лекция 1-1 – *Архитектура ОС. Микроядерная архитектура* .

17. Управление памятью в ОС. Виртуализация памяти. Основные типы распределения памяти.

- **Виртуализация памяти** — механизм, позволяющий программе видеть больше памяти, чем физически доступно.
- **Типы распределения памяти** :
- Сплошное.
- Сегментное.
- Страничное.
- Сегментно-страничное.

Источник: Лекция 2-1 – *Синхронизация* ; Лекция 2 – *Управление процессами* .

18. Управление памятью в ОС. Страничное распределение памяти.

- **Страничная организация** делит память на равные блоки — страницы.
- Преимущество: упрощает управление памятью и минимизирует фрагментацию.
- Используется таблица страниц (page table) для отображения виртуальных адресов на физические.

Источник: Лекция 2-1 – *Синхронизация* .

19. Управление памятью в ОС. Сегментное распределение памяти.

- **Сегментное распределение** делит память на сегменты различной длины.
- Каждый сегмент представляет смысловую часть программы (например, код, данные, стек).
- Удобно для защиты и управления памятью, но сложнее в реализации.

Источник: Лекция 2-1 – *Синхронизация* .

20. Управление памятью в ОС. Сегментно-страничное распределение памяти.

- Комбинированный подход: программа делится на сегменты, каждый из которых состоит из страниц.
- Обеспечивает гибкость сегментирования и простоту страничного управления.

Источник: Лекция 2-1 – *Синхронизация* .

22. Операционная система DOS. Основные команды. Командные файлы.

- **DOS** — однозадачная, однопользовательская система.

VER – выводится на экран дисплея номер версии операционной системы, которая загружается при запуске компьютера.

DIR – выводит на дисплей список файлов каталога.

TYPE – выводит на экран дисплея содержимое текстового файла.

TIME – выводит на экран дисплея текущее системное время.

DATE – отображает используемую MS DOS текущую системную дату.

MD (MKDIR) – создает новые каталоги (или подкаталоги).

CD – позволяет сменить текущий каталог.

RD – позволяет удалять подкаталог.

COPY – позволяет копировать файлы или группы файлов.

REN (RENAME) – позволяет переименовывать отдельные файлы или группы файлов.

DEL (ERASE) – позволяет удалять файлы.

TREE – выводит на экран дисплея все каталоги или их иерархическую структуру.

23. Структура диска в файловой системе DOS.

Относительная нумерация секторов жестких дисков ведется в пределах логических дисков (C:, D:, E: и т.д.), называемых **томом**. Для дискет понятия логического и физического диска совпадают.

Самый первый сектор жесткого диска (сектор 1 стороны 0 цилиндра 0) содержит главную загрузочную запись (Master boot), занимающую ровно один сектор и включающую в себя часть программы начальной загрузки и таблицу разделов диска. В таблице разделов указываются адреса и размеры разделов, на которые можно разбить жесткий диск. В MS-DOS на диске можно создать только два раздела, принадлежащих DOS: первичный и расширенный. Первичный раздел эквивалентен логическому диску C:; он используется для начальной загрузки. Расширенный раздел DOS можно разделить на тома, действующие как логические дисководы. Каждому тому дается имя дисковода от D: до Z:

и с точки зрения пользователя каждый том представляет собой независимый диск с собственной системной областью и деревом каталогов.

В MS-DOS последних версий размер первичного раздела не ограничивается; он может занимать весь диск независимо от его емкости. Рекомендуется иметь на диске только один раздел, в котором для удобства можно завести до 24 томов (логических дисководов). Жесткий диск разбивается на логические диски (C:, D:, E:, и т.д.), каждый из которых занимает целое число цилиндров; диск C: обычно начинается с начала первой свободной дорожки после главной загрузочной записи, т.е. с сектора 1 стороны 1—цилиндра 0.

Файловые системы. Файловая система FAT.

- **FAT** (File Allocation Table – таблица размещения файлов) — таблица, хранящая информацию о расположении файлов на диске.
- Версии: FAT12, FAT16, FAT32.

Источник: Лекция 3 – *Файловые системы*.

25. Назначение и структура основных областей файловой системы FAT.

- **Загрузочный сектор** — содержит информацию о диске и загрузчик.
- **Таблица FAT** — информация о свободных и занятых кластерах.
- **Корневой каталог** — список файлов верхнего уровня.
- **Область данных** — место хранения файлов.

Источник: Лекция 3 – *Файловые системы*.

26. Файловые системы. Файловая система NTFS

NTFS (New Technology File System) — это файловая система, разработанная Microsoft для использования в операционных системах Windows NT и более новых версиях (Windows XP, Vista, 7, 8, 10, 11).

- Предназначена для обеспечения **высокой надежности**, **производительности**, **безопасности** и **совместимости**.
- Поддерживает большие объемы дисков (до нескольких терабайт), длинные имена файлов, шифрование и журнал изменений.

27. Файловая система NTFS

NTFS является **расширенной файловой системой**, которая:

- Используется как стандартная файловая система в современных ОС Windows.
- Обеспечивает **логическое представление диска**, организует хранение данных и метаданных.
- В отличие от FAT, NTFS устойчива к сбоям и поддерживает восстановление данных после непредвиденного выключения.

28. Основные особенности NTFS

1. **Поддержка больших томов** – до 256 ТБ.
2. **Длинные имена файлов** – до 255 символов.
3. **Права доступа (ACL)** – возможность тонкой настройки прав пользователей.
4. **Шифрование (EFS)** – поддержка шифрования файлов и папок.
5. **Журналирование (логирование)** – запись всех операций в журнал для восстановления.
6. **Поддержка сжатия** – файлы могут быть сжаты без потери качества.
7. **Устойчивость к сбоям** – автоматическое восстановление структур при перезагрузке.
8. **Точки повторного входа (reparse points)** – поддержка символических ссылок и точек подключения.

29. Структура тома с файловой системой NTFS

Физический диск делится на **тома (логические разделы)** . Каждый том NTFS имеет следующую структуру:

- **Загрузочный сектор (Boot Sector)** – содержит информацию о структуре тома и загрузчик.
- **Master File Table (MFT)** – главная таблица файлов, где хранятся все данные о файлах и каталогах.
- **Площадка MFT (MFT Zone)** – область вокруг MFT, зарезервированная для ее роста.
- **Область данных** – место, где физически хранятся данные файлов.
- **Служебные файлы NTFS** – системные файлы, такие как \$MFT, \$LogFile, \$Bitmap, \$Volume и другие.

30. MFT (Master File Table) в NTFS

- **MFT** – центральная структура NTFS, представляющая собой базу данных всех файлов и каталогов на томе.
- Каждому файлу или каталогу соответствует **запись в MFT** .
- Записи содержат **атрибуты** , такие как имя, размер, дата создания, права доступа и указатель на расположение данных на диске.
- Первая запись MFT относится к самой таблице MFT (\$MFT).

31. Представление файлов в NTFS

- Файлы представлены как **объекты в MFT** .
- Каждый файл имеет **уникальный номер (File Reference Number)** .
- Файл может иметь несколько атрибутов: имя, содержимое, даты изменения и т.д.
- Если данные маленькие, они могут храниться прямо в MFT (резидентное хранение).
- Большие файлы хранятся вне MFT, а MFT содержит только адреса их блоков (нерезидентное хранение).

32. Атрибуты файлов в NTFS

Атрибуты — это поля в записи MFT, описывающие файл:

- **Имя файла**
- **Размер**
- **Время создания, изменения, последнего доступа**
- **Права доступа (ACL)**
- **Расположение данных на диске**
- **Флаги (архивный, скрытый, системный и т.д.)**
- **Альтернативные потоки данных (Alternate Data Streams)**

33. Организация каталогов в NTFS

- Каталоги — это специальные файлы, содержащие список других файлов и подкаталогов.
- Каждый каталог также имеет запись в MFT.

- Содержит информацию о:
- Имя элемента
- Указатель на его запись в MFT
- Тип объекта (файл или каталог)
- Для эффективного поиска используется индексная структура B+ дерево.

34. Ограничение доступа к файлам и каталогам в NTFS

- NTFS поддерживает **списки контроля доступа (ACL)** .
- ACL определяет, какие **пользователи или группы** имеют право:
- Чтение
- Запись
- Выполнение
- Полный доступ
- Можно настраивать как **разрешения** , так и **запреты** .
- Поддерживается **наследование прав** от родительских каталогов.

35. Средства самовосстановления в NTFS

- **Журнал транзакций (Log file)** – записывает все изменения перед их применением.
- При сбое ОС использует журнал для восстановления структур файловой системы.
- **Проверка целостности MFT** и других ключевых структур.
- **Автоматическое восстановление** при загрузке системы.
- Это позволяет избежать повреждения данных даже при внезапном отключении питания.

36. Недостатки файловой системы NTFS

1. **Ограниченная совместимость** – не поддерживается напрямую многими ОС (например, macOS, Android, старыми версиями Linux).
2. **Сложность восстановления** – при повреждении MFT сложно восстановить данные без специальных инструментов.
3. **Высокая нагрузка на процессор** – из-за журналирования и защиты.
4. **Не рекомендуется для флеш-накопителей** – из-за большого количества операций чтения/записи, которые уменьшают срок службы флеш-памяти.

37. Файловая система UFS (UNIX). Распределение дискового пространства в ОС UNIX

UFS (Unix File System) — классическая файловая система UNIX.

Основные элементы распределения дискового пространства:

- **Суперблок** – содержит информацию о структуре файловой системы.
- **inode'ы** – метаданные файлов (права, тип, размер, указатели на данные).
- **Блок данных** – непосредственно содержимое файлов.
- **Битовые карты (bitmap)** – показывают занятые и свободные inode и блоки.

Особенности:

- Поддерживает **жесткие и символические ссылки** .
- Разделение на **логические блоки** .

- Возможность резервирования места для root .

38. Типы файлов в UNIX

В UNIX существует несколько типов файлов:

1. **Обычные файлы** – содержат данные пользователя (текст, бинарные и т.д.).
2. **Каталоги** – специальные файлы, содержащие список имен файлов и соответствующих им inode.
3. **Символические ссылки** – указатели на другие файлы или каталоги.
4. **Жесткие ссылки** – дополнительные имена для одного и того же inode.
5. **Специальные файлы устройств** :
 - **Блочные устройства** – работа с данными блоками (например, жесткий диск).
 - **Символьные устройства** – побайтовый ввод-вывод (например, клавиатура).
1. **FIFO (очереди сообщений)** – межпроцессный обмен.
2. **Сокеты** – сетевое взаимодействие.

39. Основные команды UNIX: date, ls, less, cat, tail, du, ln, mv. Операции изменения прав доступа

date Показывает или устанавливает текущее время и дату
 ls Список файлов и каталогов
 less Просмотр содержимого файла постранично
 cat Вывод содержимого файла
 tail Вывод конца файла
 du Отображение занимаемого места
 ln Создание жесткой или символической ссылки
 mv Перемещение или переименование файла

Изменение прав доступа:

- chmod – изменяет права доступа к файлу. Пример: chmod 755 filename
- Права: r (read), w (write), x (execute)
- Группы: u (user), g (group), o (others)

40. Основные команды UNIX: cd, pwd, mkdir, rm, cp, grep, find, sort. Операции изменения владельцев файла

cd Переход в другой каталог
 pwd Показывает текущий путь
 mkdir Создание каталога
 rm Удаление файла или каталога
 cp Копирование файлов
 grep Поиск строк по шаблону
 find Поиск файлов по условиям
 sort Сортировка содержимого файла

Изменение владельца:

- chown – меняет владельца и/или группу файла. Пример: chown user:group filename

41. Структура файловой системы UNIX. Основные директории корневого каталога

Корневой каталог / содержит:

- /bin – основные исполняемые файлы.
- /boot – загрузочные файлы ядра.
- /dev – специальные файлы устройств.
- /etc – конфигурационные файлы.
- /home – домашние каталоги пользователей.
- /lib – библиотеки, необходимые для запуска программ.
- /media – точки монтирования съемных носителей.
- /mnt – временные точки монтирования.
- /opt – дополнительное программное обеспечение.
- /proc – информация о процессах и системе.
- /root – домашний каталог администратора.
- /run – временные файлы во время работы.
- /sbin – системные утилиты.
- /tmp – временные файлы.
- /usr – вторичная иерархия, программы и документация.
- /var – переменные данные (логи, очереди печати и т.д.)

42. Файл каталога. Жесткая связь между файлами. Файл символической связи

- **Файл каталога** – это список пар "имя файла → inode".
- **Жесткая связь** – создается с помощью `ln`, указывает на тот же inode, что и оригинал. Удаляется вместе с оригиналом.
- **Символическая связь (symbolic link)** – создается с помощью `ln -s`, представляет собой указатель на путь к другому файлу. Может ссылаться на удаленный файл.

43. Классы доступа и права доступа к файлам (UNIX)

Права доступа делятся на три класса:

- **User (u)** – владелец файла.
- **Group (g)** – группа владельца.
- **Others (o)** – остальные пользователи.

Права:

- **r** – чтение
- **w** – запись
- **x** – выполнение

Пример: `rw-r--r--` означает:

- Владелец: чтение и запись
- Группа: только чтение
- Все остальные: только чтение

44. Назначение реестра

Реестр Windows — это центральное хранилище параметров и настроек операционной системы и установленных программ.

- Хранит информацию:
- Настройки оборудования
- Конфигурация ОС
- Параметры учетных записей
- Информация о приложениях

45. Структура реестра

Реестр состоит из **ключей (keys)** и **параметров (values)** .

Основные разделы:

- HKEY_CLASSES_ROOT – связь между файлами и программами.
- HKEY_CURRENT_USER – настройки текущего пользователя.
- HKEY_LOCAL_MACHINE – настройки всей системы.
- HKEY_USERS – параметры всех пользователей.
- HKEY_CURRENT_CONFIG – текущая аппаратная конфигурация.

46. Хранение данных реестра

- Реестр хранится в виде **файлов баз данных** в папке %SystemRoot%\System32\config.
- Эти файлы являются **скрытыми системными файлами** .
- При загрузке Windows загружает эти файлы в память и предоставляет к ним доступ через интерфейс редактора реестра (regedit).

47. Программа Regedit

Regedit.exe — это **редактор реестра Windows** , позволяющий:

- Просматривать и редактировать ключи и значения реестра.
- Импортировать и экспортировать части реестра.
- Поиск по реестру.