**26. Средства автоматической настройки в ОС Windows.**

* Командное консольное окно.
* BAT-файлы.
* PowerShell.
* WMI.
* ETW.
* Групповые политики.
* Административные шаблоны.
* Файловые системы.
* Динамические диски.
* Управление сжатием на дисках NTFS.
* Шифрование файлов.
* Управление дисковыми квотами.
* Использование Консоли восстановления.

**Командное консольное окно**

В Windows есть две оболочки командной строки: командная оболочка **cmd** и **PowerShell**.

Каждая оболочка — это программная программа, которая обеспечивает прямую связь между оператором и операционной системой или приложением, предоставляя среду для автоматизации ИТ-операций.

Командная оболочка **cmd** была первой оболочкой, встроенной в Windows, для

автоматизации повседневных задач, с пакетными (.bat или .cmd) файлами.

**PowerShell** был разработан для расширения возможностей командной оболочки для выполнения команд PowerShell, называемых **командлетами**. Командлеты похожи на команды Windows, но предоставляют более расширяемый язык сценариев.

Команды Windows и командлеты PowerShell можно запускать в PowerShell, но командная оболочка может выполнять только команды Windows, а не командлеты PowerShell.

(также тут можно написать, как открыть cmd или PowerShell, а также написать примеры команд)

**BAT-файлы**

BAT-файл (Batch-файл)текстовый файл, содержащий команды системы, которые могут выполняться последовательно. Расширение .cmd или .bat.

Комментарии начинаются с REM (remark)

Строки, начинающиеся с @, не выводятся на экран.

Допускаются программные конструкции, например:

**for** %%i **in** (1 1 5) **do** *echo* %%i (%% переменные в циклах, а % переменные вне цикла) ( **in** (1-нач знач 1-шаг 5-конечное знач)

**if** a *equ* 0 *echo* 0 **else** *echo* notzero (*equ- =)*

Для вызова другого скрипта используется команда call

Для создания переменной используется команда set (set a=0)

**PowerShell** — расширяемое средство автоматизации от Microsoft с открытым исходным кодом.(выпущен августе 2016 года.)

* Основан на .NET Framework и интегрирован с ним
* Интеграция с COM (Component Object Model), WMI(Windows Management Instrumentation) и ADSI (Active Directory Service Interfaces)
* Командлеты
* доступ к источникам данных, таким как файловая система или реестр Windows
* имеет собственную расширяемую справку, доступную через **командлет** **Get-Help.**

**WMI**

Инструментарий управления Windows **(WMI)** — это инфраструктура для управления данными и операциями в операционных системах Windows. Он представляет собой реализацию стандарта WBEM, ориентированного на создание универсального интерфейса мониторинга и управления различными системами.

**В основе** структуры данных в WBEM **лежит Common Information Model (CIM),**

реализующая ООП-подход к представлению компонентов системы.

**wbemtest.exe** — графическая утилита для взаимодействия со структурой WMI на

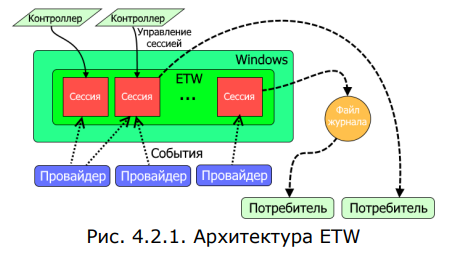
локальном или удаленном компьютере.

**wmic.exe** — консольная утилита для взаимодействия со структурой WMI на локальном или удаленном компьютере.

**Event Tracing for Windows (ETW)** — это служба, которая позволяет получать события от одного или нескольких поставщиков событий в режиме реального времени или из файла \*.etl за некоторый временной период.

Архитектура ETW включает в себя 4 элемента (рис. 4.2.1):

* поставщики событий (providers)
* потребители событий (consumers)
* контроллеры ETW (controllers)
* сессии ETW (event tracing sessions)



**Групповая политика** — важный элемент любой среды Microsoft Active Directory (AD). Её основная **цель** — дать ИТ-администраторам возможность централизованно управлять пользователями и компьютерами в домене.

Групповая политика **состоит из** набора политик, называемых **объектами групповой политики (GPO)**.

При создании домена AD автоматически создаются два объекта групповой политики:

• **политика домена** по умолчанию устанавливает базовые параметры для всех пользователей и компьютеров в домене в трех плоскостях: политика паролей, политика блокировки учетных записей и политика Kerberos;

• **политика контроллеров домена** по умолчанию устанавливает базовые параметры безопасности и аудита для всех контроллеров домена в рамках домена

Объект групповой политики не действует, пока не будет связан с контейнером Active

Directory, например, сайтом, доменом или подразделением.

**Административные** шаблоны (файлы) в формате XML, написанные на различных языках, которые задают параметры групповых политик, основанные на значениях реестра. Эти параметры отображаются и редактируются в локальном редакторе групповых политик на компьютере. Административные шаблоны позволяют администраторам устанавливать и изменять настройки безопасности, поведения и другие параметры на компьютере или сети через групповые политики.

**Два вида административных шаблонов:**

* **ADMX** — не зависящий от языка файл установки, который указывает количество и тип параметров политики, расположение по категориям согласно отображению файла в редакторе локальных групповых политик.
* **ADML** — файл установки на определенном языке, который предоставляет

связанные с языком сведения для ADMX-файла. позволяет параметру политики отображаться в редакторе локальных групповых политик на нужном языке. Вы можете добавлять новые языки, добавляя новые ADML-файлы на нужных языках.

**Файловая система** – это способ организации и хранения файлов на жестком диске

компьютера.

Основным типов является **NTFS** (New Technology File System). NTFS обладает высокой степенью надежности, поддерживает шифрование(технологии EFS (Encrypting File System)) и права доступа к файлам и папкам. Кроме того, NTFS позволяет работать с файлами размером до 16 терабайт. (стоит по умолч в Windows)

Другим типом явл **FAT** (File Allocation Table). FAT поддерживает работу с файлами менее 4 гигабайт и не обеспечивает такую высокую степень надежности, как NTFS. Однако FAT более совместима с различными операционными системами и устройствами.

**Динамические диски** предоставляют функции, которых нет у базовых дисков. Вы можете создавать тома, охватывающие несколько дисков, и отказоустойчивые тома.

На каждом динамическом диске компьютера хранится копия базы данных динамического диска.

Следующие операции можно выполнять только на динамических дисках:

* Создание и удаление составных, чередующихся и зеркальных томов.
* Расширьте простой том до несмежного пространства или составного тома.
* Удалите зеркало из зеркального тома.
* Восстановление зеркальных томов.
* Повторно активируйте отсутствующий или отключенный диск

Особенности:

* Вы не можете преобразовать базовый диск в динамический диск, если на диске

не имеется хотя бы 1 МБ неиспользуемого пространства из-за базы данных

диспетчера логических дисков.

* Вы не можете преобразовать динамический диск в базовый без потери данных.
* Вы не можете использовать Windows PowerShell для управления динамическими дисками

**Управление сжатием на дисках NTFS**

два типа сжатия:

* NTFS-сжатие на уровне файловой системы,
* Сжатые ZIP-папки (Compressed Folders).

**Сжатие NTFS** – это архивирование на уровне файловой системы NTFS, выполняется оно прозрачно драйвером файловой системы. NTFS сжатие может применяться к файлам, папкам и дискам целиком. Для этого каждому сжатому объекту присваивается специальный атрибут сжатия (compression state), который указывает, сжат файл или нет.

* Сжатие NTFS возможно только на разделах с файловой системы NTFS( Zip и в NTFS и в FAT)
* При доступе к сжатому файлу или папке осуществляется прозрачная декомпрессия, т.е. пользователь не видит различий между сжатыми и обычными файлами
* Скорость доступа к сжатому файлу ниже, т.к. систем требуется некоторое время на его распаковку
* Сжатие более слабое, чем при использовании ZIP-папок, но скорость его выполнения гораздо выше

**Шифрование файлов**

Windows поддерживает две технологии шифрования:

* **EFS** — на уровне файлов и папок;
* **BitLocker** — на уровне томов.

Encrypting File System (**EFS**) — система шифрования данных, реализующая шифрование на уровне файлов (кроме «домашних» версий Windows). Предоставляет возможность «прозрачного шифрования»,т.е

EFS работает, шифруя каждый файл с помощью алгоритма симметричного шифрования, зависящего от версии операционной системы и настроек. При этом используется случайно сгенерированный ключ для каждого файла, называемый File Encryption Key (FEK)

**BitLocker** — это функция безопасности Windows, которая обеспечивает шифрование для целых томов, устраняя угрозы кражи или раскрытия данных с потерянных, украденных или неправильно списанных устройств. TPM работает с BitLocker, чтобы убедиться, что устройство не было изменено, пока система находится в автономном режиме.

**Дисковые квоты** позволяют контролировать сколько места используют пользователи на файловой системе серверов и рабочих станций.

ОС Windows Server поддерживает два типа квотирования:

* квотирование на базе File Server Resource Manager (дисковые квоты FSRM);
* NTFS квоты.

С помощью дисковых квот Windows вы можете ограничить максимальный размер файлов и папок каждого пользователя так, чтобы он не превысил установленного лимита и не занял своими данными весь диск.

**Использование консоли восстановления**

Среда восстановления Windows 10 позволяет производить операции, имеющие отношение к восстановлению работоспособности системы: использовать **особые варианты загрузк**и (например, безопасный режим или отключение проверки цифровой подписи драйверов), **выполнять сброс ОС** или **автоматическое восстановление загрузчика Windows 10** и т.д.

**27. Основные понятия системы UNIX. Пользователи системы, атрибуты пользователя. Файловая структура ОС.**

ОС UNIX базируется на работе с двумя основными объектами - **файлами** и **процессами**.

**Файлы** хранят данные пользователя, а **процессы** определяют функциональность системы.

Работа с файлами осуществляется через чтение и запись специальных файлов, а выполнение программ включает считывание исполняемого кода из файла в память.

Операционная система UNIX организована как пирамида, где аппаратное обеспечение находится в основании, а операционная система управляет аппаратурой и предоставляет интерфейс системных вызовов для программ.

Программы используют системные вызовы для создания и управления процессами, файлами и другими ресурсами.

Графический интерфейс пользователя основан на оконной системе X, которая управляет визуальным представлением на экране.

Операционная система также включает стандартные программы, такие как командный процессор, компиляторы, редакторы, и поддерживает графические интерфейсы пользователя через X Windowing System.

**Пользователи системы**

Прежде чем клиент сможет начать работу с ОС UNIX, он должен стать пользователем системы, т.е. получить имя, пароль и ряд других атрибутов **Пользователем** является объект, который обладает определенными правами и может запускать на выполнение программы и владеть файлами. Пользователями могут быть отдельные клиенты, удаленные компьютеры или группы пользователей с одинаковыми правами и функциями. В системе существует один пользователь, обладающий неограниченными правами это **суперпользователь** или **администратор** системы (обычно с именем **root**).

Система различает пользователей по идентификатору пользователя UID.

**Группа** — список пользователей, имеющих сходные задачи. Каждая группа имеет уникальное имя, а система различает группы по групповому идентификатору (GID).Информация о пользователях обычно хранится в специальном файле: */etc/passwd,* о группах *— /etc/group*. (файлы доступены только для чтения, писать может только админ)

**Атрибуты пользователя**

все атрибуты пользователя хранятся в файле /etc/passwd

Каждая строка файла является записью конкретного пользователя и имеет следующий формат:

name:passwd-encod:UID:GID:comments:home-dir:shell

всего семь полей (атрибутов), разделенных двоеточиями.

* name: Регистрационное имя пользователя, используемое при входе в систему.
* passwd-encod: Закодированный пароль пользователя. Алгоритм кодирования предотвращает декодирование пароля. В некоторых системах пароль хранится в отдельном файле, а в поле passwd-encod может быть символ 'х' или '!'.
* UID: Идентификатор пользователя, внутреннее представление пользователя в системе. Суперпользователь (администратор) имеет UID=0.
* GID: Идентификатор первичной группы пользователя, соответствует идентификатору в файле /etc/group.
* comments: Полное "реальное" имя пользователя или дополнительная информация.
* home-dir: Домашний каталог пользователя, место, где пользователь оказывается при входе в систему.
* shell: Имя программы, используемой в качестве командного интерпретатора при входе пользователя. Обычно это /bin/sh, /bin/csh или /bin/ksh.

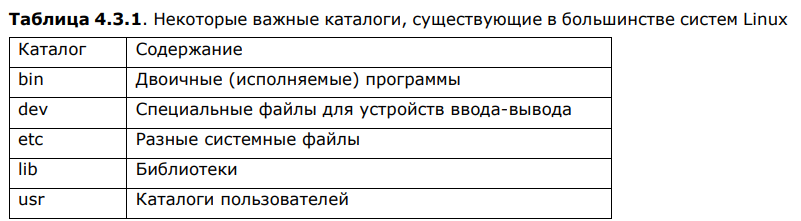
**Файловая структура ОС.**

В начале развития Linux использовалась файловая система MINIX 1, но ограничения по длине имен и размеру файлов привели к разработке улучшенной файловой системы ext. Затем появилась файловая система **ext2**, ставшая основной для Linux.

**Файл в системе Linux** — это последовательность байтов произвольной длины (от 0 до некоторого максимума), содержащая произвольную информацию.

Не делается различия между текстовыми (ASCII) файлами, двоичными файлами и любыми другими типами файлов. Значение битов в файле целиком определяется владельцем файла. Имена файлов ограничены 255 символами. В именах файлов разрешается использовать все ASCII-символы, кроме символа NULL.

Для удобства файлы могут группироваться в **каталоги**. Каталоги хранятся на диске в виде файлов. Каталоги могут содержать подкаталоги, что приводит к иерархической файловой системе. Корневой каталог называется **/**



Существует два способа указания имени файла в системе Linux: абсолютный путь, начинающийся от корневого каталога, и относительный путь, указывающий относительно текущего рабочего каталога.

Стандарт POSIX обеспечивает гибкий механизм блокировки файлов, позволяя процессам блокировать как единичные байты, так и целые файлы за одну неделимую операцию. Стандарт определяет два типа блокировок: с монополизацией (эксклюзивные) и без монополизации (общие). Если часть файла имеет блокировку без монополизации, то можно повторно установить такую же блокировку, но блокировка с монополизацией будет отвергнута. Если область файла заблокирована с монополизацией, то любые попытки блокировки этой области будут отвергнуты до снятия блокировки. Успешная установка блокировки требует доступности каждого байта в заблокированной области.

**28. ОС UNIX: особенности процессов, сигналы, обработка сигналов**.

Основными активными сущностями в системе Linux являются **процессы**. Каждый процесс выполняет одну программу и изначально получает один поток управления. Иначе говоря, у процесса есть один счетчик команд, который отслеживает следующую исполняемую команду. Linux позволяет процессу создавать дополнительные потоки (после того, как он начинает выполнение).

Системный вызов **fork** создает точную копию исходного процесса, называемого родительским процессом (parent process). У родительского и у дочернего процессов есть собственные (приватные) образы памяти. Если родительский процесс впоследствии изменяет какие-либо свои переменные, то эти изменения остаются невидимыми для дочернего процесса (и наоборот).

**fork** возвращает дочернему процессу число 0, а родительскому — отличный от нуля PID дочернего процесса. **getpid**

Типы процессоров:

**Системные процессы** - часть ядра, всегда в оперативной памяти. Не имеют соответствующих исполняемых файлов, запускаются особым образом при инициализации ядра. Примеры: shed, vhand, bdfflush, kmadaemon, и init.

**Демоны (daemons)** фоновые процессы, работающие в фоновом режиме и обслуживающие различные системные задачи или услуги. Они часто запускаются при загрузке системы и работают независимо от активности конкретного пользователя. Типичным демоном является ***cron***. Он просыпается раз в минуту, проверяя, не нужно ли ему что-то сделать. Есть работа-выполняет, нету-спит до след проверки. Позволяет планировать в системе Linux активность на минуты, часы, дни и месяцы вперед.

**Прикладные процессы** - все остальные, выполняющиеся в системе, часто порождаемые пользовательским сеансом. Пример: команды, запущенные пользователем. Время их выполнения ограничено сеансом работы.

**Сигналы** — это способ информирования процесса со стороны ядра о происшествии некоторого события. Один процесс может посылать сигнал другому через системный вызов kill.

**Обработка сигналов** в Linux осуществляется через установку обработчиков сигналов. Когда процесс получает сигнал, ядро отправляет управление к обработчику, который был предварительно установлен для этого сигнала. Процесс может выбрать различные действия при получении сигнала:

**Игнорирование (SIG\_IGN):** Процесс может проигнорировать сигнал, если установлен обработчик SIG\_IGN (ignore).

**По умолчанию (SIG\_DFL)**: Если не установлен пользовательский обработчик, применяется действие по умолчанию (default). Например завершение процесса.

**Пользовательский обработчик (функция):** Процесс может установить свою собственную функцию-обработчик. Когда сигнал поступает, управление передается этой функции.

**29. Основные принципы функционирования Linux. Основные компоненты Linux. Дистрибутивы Linux. Файловая система Linux**.

Linux имеет низкие аппаратные требования в текстовом режиме, что позволяет работать на машинах с 486 процессором и 16 MB RAM. Он организован по принципу клиент-сервер, где компьютеры в сети взаимодействуют, создавая единый компьютер. Пользователи работают через виртуальные терминалы, переключаясь<Alt> + <F1...F12>. Один компьютер может одновременно выполнять роль сервера и рабочей станции.

Linux - это ядро и ядро каждого дистрибутива Linux. Программное обеспечение ядра Linux поддерживается группой людей, возглавляемой Линусом Торвальдсом.

Версию используемого ядра Linux можно узнать c помощью команды **uname**

ОС Linux обладает различными компонентами и возможностями:

* *Планировщик*: Управляет приоритетами процессов.
* *Файл подкачки:* Используется для хранения неактивных процессов на диске при переполнении оперативной памяти.
* *Модули:* Позволяют динамически добавлять поддержку новых устройств, что помогает сократить размер ядра.
* *Файловые системы:* Поддерживают различные типы файловых систем, включая ext3, ReiserFS, VFAT и NTFS.
* *Механизмы защиты:* Используются для установки разрешений на чтение, запись и выполнение файлов, обеспечивая безопасность.
* *Инструменты администрирования*: управления пользователями, дисками, сетями и другими ресурсами.
* *Серверные возможности:* Позволяют использовать Linux в роли сервера, предоставляя различные службы, такие как веб-серверы, почтовые серверы и другие.
* *Система управления пакетами:* Позволяет автоматизировать установку, обновление и удаление программного обеспечения, используя различные форматы пакетов, такие как RPM, DEB, и другие.

Мы можем распределить **дистрибутивы Linux** на три группы:

***Enterprise Grade Linux*** Дистрибутивы этой группы предназначены для развертывания в крупных организациях с использованием оборудования предприятия.( Red Hat Enterprise Linux, CentOS, SUSE Linux Enterprise Server, Debian GNU/Linux , Ubuntu LTS)

Особенности:

* Стабильность и долгосрочная поддержка.
* Ориентированы на надежность и безопасность.
* Медленное внедрение новых технологий.
* Предоставляют более старые, но стабильные версии программного обеспечения.

***Consumer Grade Linux*** Дистрибутивы этой группы больше ориентированы на малый бизнес или домашних пользователей и любителей.( Fedora, Ubuntu non-LTS, openSUSE)

Особенности:

* Ориентированы на новое оборудование и современные драйверы.
* Регулярные выпуски с обновлениями.
* Более быстрое внедрение новых технологий.

***Experimental and Hacker Linux*** Дистрибутивы этой группы используют самые современные технологии. (Arch, Gentoo)

Особенности:

* Использование самых последних технологий.
* Скользящая модель выпуска.
* Предоставляют возможность для тестирования будущих функций.
* Разработаны для пользователей, готовых к возможным проблемам и изменениям.

**Файловая система** в ОС Linux как и в большинстве других систем имеет иерархическую (древовидную) структуру. Все объекты являются файлами, в том числе и директории для организации доступа к файлам.

Существуют следующие типы файлов: обычные файлы, каталоги, символьные ссылки, блочные устройства, символьные устройства, сокеты, каналы. Тип выводится **ls -l.**

всегда есть только один корневой каталог, который называется /

Абсолютные и относительные пути

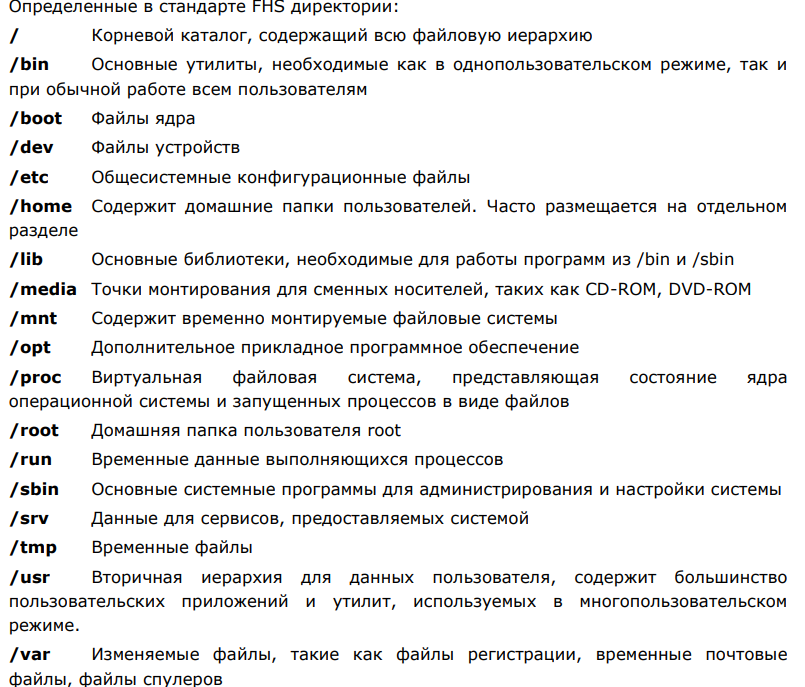
Имя может содержать любые символы (в том числе и кириллицу), кроме / ? < > \* " | Максимальная длина имени файла — 254 символа

Точка **.** в начале имени файла делает его **скрытым**, то есть, он не показывается в выводе команды ls.

Команда **pwd** (Print Working Directory) возвращает полный путь к текущей директории командной оболочки (хранится в переменной PWD) Для смены текущей директории используется команда **cd** (Change Directory)

**Домашний каталог (домашняя папка, домашняя директория)** – предназначен для хранения собственных данных пользователя Linux и личных настроек для программ. Полный путь к домашнему каталогу хранится в переменной окружения **HOME**, Для обычных пользователей домашний каталог находится в директории **/home**

**FHS (Filesystem Hierarchy Standard)** – стандарт файловой системы. В FHS все файлы и каталоги находятся внутри корневого каталога, даже если они расположены на различных физических носителях. В корневой директории должны быть следующие директории:



**30. ОС Linux: управление процессами, выполнение задач в фоновом режиме, изменение приоритетов выполняющихся программ.**

Процессы существуют в иерархии: после загрузки ядра в память запускается первый процесс (init или systemd), который, в свою очередь, запускает другие процессы, которые, опять же, могут запускать другие процессы.

Каждый раз, когда пользователь вводит команду, запускается программа и генерируется один или несколько процессов. Каждый процесс имеет уникальный идентификатор (**PID**) и идентификатор родительского процесса (**PPID**).

Команда **top** динамически отображает все запущенные процессы

Иерархию процессов команда **pstree**.

Завершить процесс **kill**.

Для выполнения команды в **фоновом режиме** достаточно добавить в конце символ амперсанда **&**.

Работая в фоновом режиме, команда все равно продолжает выводить сообщения в терминал, из которого была запущена. Для этого она использует потоки stdout и stderr, которые можно закрыть при помощи следующего синтаксиса:

$ command > /dev/null 2>&1 &

Узнать состояние всех остановленных и выполняемых в фоновом режиме задач в рамках текущей сессии терминала можно при помощи утилиты **jobs** **-l.**

В любое время можно вернуть процесс из фонового режима на передний план **fg**. Если в фоновом режиме выполняется несколько программ, следует указывать номер. Например: **$ fg %1**

Для завершения фонового процесса применяют команду **kill** с номером программы

Утилита **nice** — программа, предназначенная для запуска процессов с изменённым приоритетом nice.

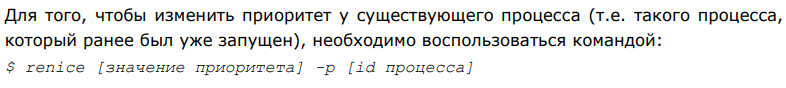
Приоритет **nice** (целое число) процесса используется планировщиком процессов ядра ОС при распределении процессорного времени между процессам

nice – значение любезности, чем меньше, тем выше приоритет

значение nice – минимальное значение приоритета =лучшее значение = самый высокий приоритет.

nice – диапазон приоритетов [-20, 19], default = 0;

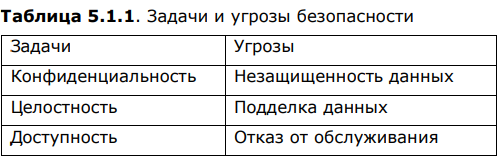
Чтобы установить значение nice ниже нуля, требуются права суперпользователя



**31. Понятие безопасности ОС. Основные угрозы безопасности ОС. Методы и защитные механизмы операционных систем.**

Понятие безопасности операционной системы (ОС):

**Безопасность** в контексте операционных систем охватывает механизмы и стратегии, направленные на защиту информации и обеспечение нормальной работы системы. Для разграничения, термин "безопасность" будет использоваться для общих вопросов, а "защитные механизмы" — для конкретных мер, используемых ОС.



Первое свойство — конфиденциальность (confidentiality) — направлено на сохранение секретности данных

Второе свойство — целостность (integrity) — означает, что пользователи, не обладающие необходимыми правами, не должны иметь возможности изменять какие-либо данные без разрешения их владельцев.

Третье свойство — доступность (availability) — означает, что никто не может нарушить работу системы и вывести ее из строя

Классификация угроз **по цели** атаки:

• несанкционированное чтение информации;

• изменение информации;

• уничтожение информации;

• полное или частичное разрушение операционной системы.

Классификация угроз **по принципу воздействия** на операционную систему:

• использование известных (легальных) каналов получения информации

• использование скрытых каналов получения информации

• создание новых каналов получения информации с помощью программных закладок.

Классификация угроз **по характеру** воздействия на ОС:

• активное воздействие – несанкционированные действия злоумышленника в системе;

• пассивное воздействие – несанкционированное наблюдение злоумышленника за процессами, происходящими в системе

Операционная система **может подвергнуться следующим типичным атакам**:

• *сканирование файловой системы.* Злоумышленник просматривает файловую систему компьютера и пытается прочесть (или скопировать) все файлы подряд.

• *подбор пароля.* несколько методов подбора паролей:

* тотальный перебор;
* тотальный перебор, оптимизированный по статистике встречаемости символов или с помощью словарей;
* подбор пароля с использованием знаний о пользователе (его имени, фамилии, даты рождения, номера телефона и т. д.);

• *кража ключевой информации*. Носитель с ключевой информацией (смарткарта, Touch Memory и т. д.) может быть просто украден;

• *сборка мусора*. Во многих операционных системах удаленная пользователем информация не физически уничтожается, а лишь помечается как удаленная, что предоставляет возможность злоумышленникам восстановить и просмотреть данные.

• *превышение полномочий*. Злоумышленник, используя ошибки в программном обеспечении ОС получает полномочия, превышающие те, которые ему предоставлены в соответствии с политикой безопасности.

• *программные закладки.*

• *жадные программы* – это программы, преднамеренно захватывающие значительную часть ресурсов компьютера, в результате чего другие программы не могут выполняться или выполняются крайне медленно. Запуск жадной программы может привести к краху операционной системы.

Существуют ***два основных подхода*** *к созданию защищенных операционных систем*: **фрагментарный**, где защитные функции добавляются отдельно после создания основной системы, и **комплексный**, где защитные функции внедряются в архитектуру системы на этапе проектирования, обеспечивая их тесное взаимодействие и интеграцию.

Основные административные меры защит.

• Постоянный контроль корректности функционирования ОС, особенно ее подсистемы защиты

• Организация и поддержание адекватной политики безопасности.

•Осведомление пользователей о необходимости соблюдения мер безопасности при работе с ОС и контроль за соблюдением этих мер.

• Регулярное создание и обновление резервных копий программ и данных ОС.

• Постоянный контроль изменений в конфигурационных данных и политике безопасности ОС.

**32. Механизмы безопасности в операционных системах семейства Windows.**

• Идентификаторы безопасности (SID).

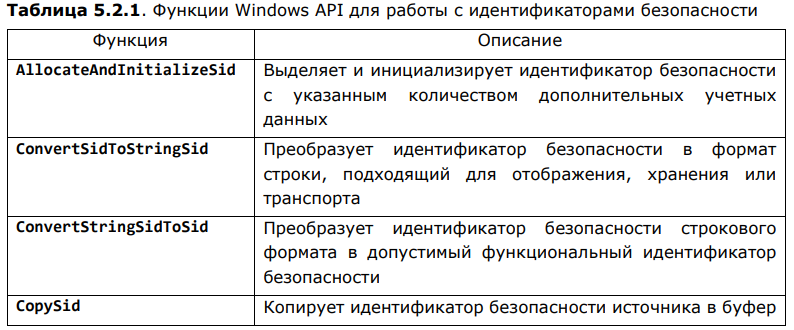
• Маркеры защиты.

• Списки управления доступом (ACL).

• Доменные службы Active Directory.

• Виды групп пользователей.

**Идентификатор безопасности (SID)** — это уникальное значение переменной длины, используемое для идентификации доверенного лица. Каждая учетная запись имеет уникальный идентификатор безопасности, выданный центром сертификации. Каждый раз, когда пользователь входит в систему, система получает идентификатор безопасности для этого пользователя из базы данных и помещает его в маркер доступа для этого пользователя.





**Маркер доступа** представляет собой объект, содержащий информацию о контексте безопасности процесса или потока. Он включает удостоверение и привилегии учетной записи пользователя, связанной с процессом или потоком. После успешной аутентификации пользователя при входе в систему, система создает маркер доступа. Каждый процесс, выполняемый от имени данного пользователя, обладает копией этого маркера доступа. Маркер доступа используется системой для идентификации пользователя при взаимодействии с защищенными объектами или при выполнении системных задач, требующих привилегий.

Маркеры доступа содержат следующие сведения:

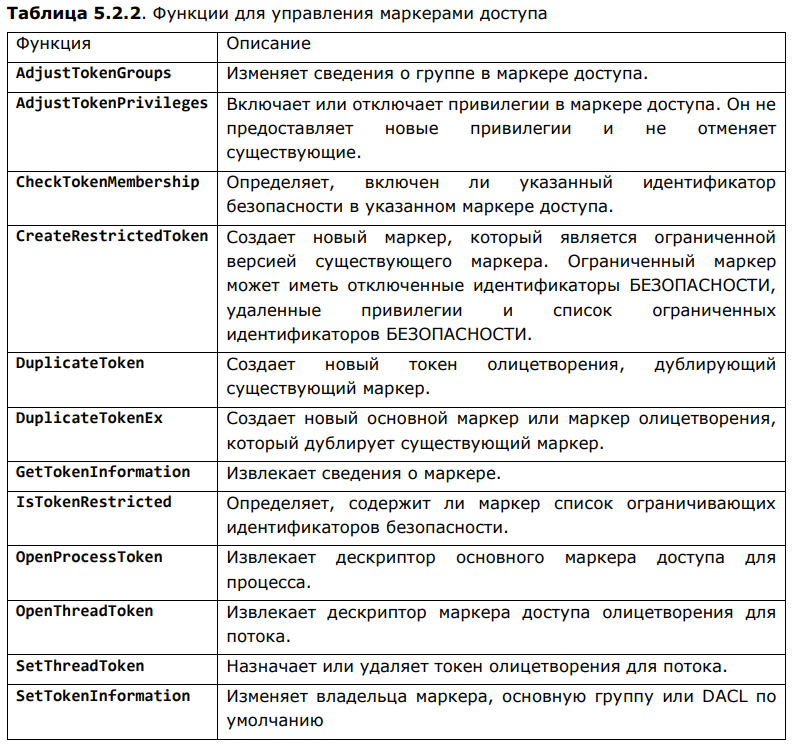
• Идентификатор безопасности (SID) для учетной записи пользователя

• Идентификаторы безопасности для групп, членом которых является пользователь

• Идентификатор безопасности входа, который идентифицирует текущий сеанс входа в систему.

• Список привилегий, которыми пользовались пользователи или группы пользователей.

• Идентификатор безопасности владельца • Идентификатор безопасности для основной группы и тд



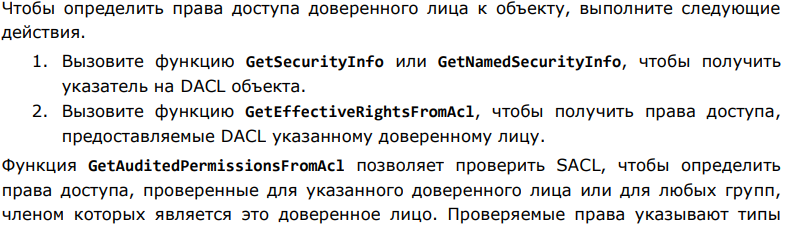
**Дескриптор безопасности** содержит сведения о безопасности, связанные с защищаемым объектом.

**Списки управления доступом (ACL)** состоят из записей управления доступом (ACE). Каждый ACE в ACL идентифицирует доверенного лица и указывает права доступа, разрешенные, запрещенные или регистрируемые для этого доверенного лица.

Дескриптор безопасности для защищаемого объекта может содержать два типа списков управления доступом: **DACL** и **SACL**

**Список управления доступом (DACL)** на уровне пользователей определяет, кому разрешен или запрещен доступ к объекту. При запросе доступа система проверяет записи ACE в DACL объекта, определяя, следует ли предоставлять доступ. Если DACL отсутствует, полный доступ предоставляется всем пользователям. Если DACL не содержит соответствующих записей ACE, доступ к объекту отклоняется. Система проверяет ACE последовательно, пока не найдет разрешающую запись или отклонит доступ.

**Системный список управления доступом (SACL)** позволяет регистрировать попытки доступа к объекту, создавая записи аудита в журнале событий безопасности. Каждая запись ACE в SACL указывает типы попыток доступа, которые приведут к созданию записи в журнале событий. SACL может фиксировать попытки доступа при успехе и/или при неудаче.



**Active Directory (AD)** является службой каталогов, предоставляющей иерархическую структуру для хранения и предоставления сведений об объектах в сети. В контексте AD DS (Active Directory Domain Services), эта служба хранит информацию о пользователях, группах, серверах и других ресурсах в сети. Администраторы и авторизованные пользователи могут получать доступ к этим данным для управления и использования в сети

Active Directory также включает следующие **компоненты**:

• *Схема* — набор правил, который определяет классы объектов и атрибутов, содержащихся в каталоге, ограничения и ограничения экземпляров этих объектов, а также формат их имен.

• *Глобальный каталог*, содержащий сведения о каждом объекте в каталоге. Это позволяет находить сведения о каталоге независимо от того, какой домен в каталоге фактически содержит данные.

• *Механизм запроса и индекса*, чтобы объекты и их свойства могли быть опубликованы и найдены сетевыми пользователями или приложениями.

• *Служба репликации*, которая синхронизирует данные каталога по сети. Все контроллеры домена в домене участвуют в репликации и содержат полную копию всех сведений о каталоге для своего домена. Любые изменения данных каталога реплицируются в домене на все контроллеры домена

(и еще дохуя инфы которую вы можете прочесть в лк 5\_2)

Active Directory имеет два типа групп:

• **Группы безопасности**. Используйте для назначения разрешений общим ресурсам.

• **Группы рассылки**: создание списков рассылки электронной почты.

Каждая группа имеет область, определяющий степень применения группы в дереве домена или лесу. Область группы определяет, где можно предоставить разрешения сети для группы.

Active Directory определяет следующие три области для групп:

• Универсальная • Глобальная • Локальная в домене

**33. Механизмы безопасности в операционных системах семейства Linux.**

В операционной системе GNU/Linux существуют следующие типы файлов:

• обычные файлы (-) все файлы с данными

• каталог (d) тип файла, данными которого является список имен других файлов и каталогов, вложенных в данный катало

• символьные ссылки (l) файл, в данных которого содержится адрес другого файла по его имени (а не индексному дескриптору).

• блочные устройства (b)

• символьные устройства (c) файлы устройств предназначены для обращения к аппаратному обеспечению компьютера (д

• сокеты (s)

• каналы (p)

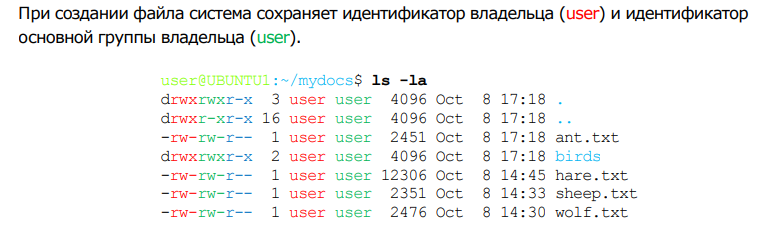
Каждый тип имеет собственное обозначение одним символом.

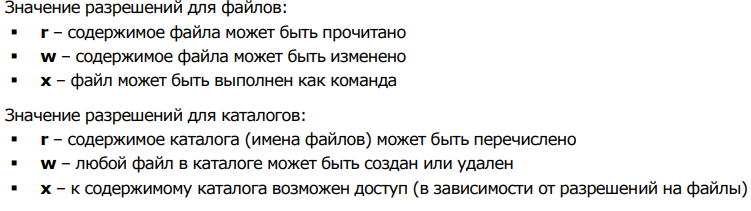
В Linux у каждого файла и каждого каталога есть **два владельца:** *пользователь* и *группа.* владельцы устанавливаются при создании файла или каталога.

Чтобы увидеть текущие назначения владельца, вы можете использовать команду

**ls-l.** Эта команда показывает пользователя и группу-владельца.

Команда **chown** позволяет сменить владельца (команда **chgrp** – только группу-владельца).





**Управление разрешениями**

Команда **chmod** позволяет изменять разрешения двумя способами:

* символьным chmod a+x cat.txt
* числовым chmod 750 animals

Некоторые опции:

-R, --recursive изменять файлы и директории рекурсивно

-v, --verbose выводить диагностику о каждом обрабатываемом файле

-c, --changes выводить диагностику только для изменяемых файлов

В операционной системе Linux базовые права для директории равны 0777 (rwxrwxrwx), а для файла 0666 (rw-rw-rw) По умолчанию umask 0002 используется для обычного пользователя. С этой маской права по умолчанию для директории равны 775, а для файла – 664.

**атрибуты файла** — это свойства метаданных, которые описывают поведение файла. Например, атрибут может указывать, сжат ли файл, или указывать, можно ли удалить файл. Вы можете просмотреть атрибуты файла с помощью команды **lsattr**: lsattr todo.txt.

Одно из распространенных применений **chattr** — установка неизменяемого флага для файла или каталога, чтобы пользователи не могли удалить или переименовать файл.

**Управление свойствами файлов**

В Linux каждый файл имеет довольно много свойств, например, права доступа устанавливаются трижды (для владельца, группы и всех прочих), метки времени также бывают трёх разных видов (время создание, доступа и изменения).

У каждого файла доступны следующие метки времени:

• Доступ

• Модифицирован

• Изменён

• Создан

Посмотреть метки времени папки можно также с помощью команды **stat**:

stat /путь/до/папки

С помощью команды **touch** можно изменить три метки времени файла или папки:

• время доступа • время модификации • время изменения статуса