**1 блок**

**1. Определение администрирования, администрирования ИС. Функции администратора ИС.**

ИС – это система, в которой присутствуют информационные процессы (хранение, передача информации) ИС, получая информацию, преобразует ее в информационный продукт.

**Администрирование** – процесс управления какой-либо системой/процедуры управления, регламентирующие некоторые процессы или части процессов.

**Администрирование ИС** – это ее инсталляция, управление доступом к ней, обеспечение целостности ИС, обеспечение ее работоспособности.

**Функции:**

* Решение проблемных ситуаций
* Управление ресурсами
* Управление конфигурацией, направленное на повышение надежности и эффективности всех компонентов ИС
* Контроль производительности
* Защита данных

**2. Основные категории пользователей ИС (различные категории). Административный персонал ИС. Стандартный пакет нормативных документов администратора ИС. (определения вопрос1)**

**Категории пользователей:**

1. Администратор ИС – это специалист (или группа специалистов), отвечающий за эксплуатацию системы и обеспечение ее работоспособности, понимающий потребности конечных пользователей, работающий с ними в тесном контакте и отвечающий за определение, загрузку, защиту и эффективность работы банка данных.
2. Прикладные программисты – занимаются разработкой программ для решения прикладных задач, реализации запросов к бд.
3. Системные программисты – осуществляют поддержку ИС и обеспечивают ее работоспособность, занимаются разработкой и сопровождением базового ПО компьютеров (ОС, систем управления бд, трансляторов, сервисных программ общего назначения).
4. Конечный пользователь (потребитель информации) – лицо или коллектив, в интересах которых работает ИС. Он работает с ИС повседневно, связан с ограниченной областью деятельности и, как правило, не является программистом. Например, бухгалтер, маркетолог, финансовый менеджер или руководитель подразделения.

**Административный персонал:**

Все аспекты деятельности всех администраторов должны быть обеспечены нормативными и методическими документами.

* Администратор 1 – оптимизация настроек, мониторинг производительности, модернизация, тех. обслуживание и профилактика, резервное копирование
* Администратор 2 – регистрация новых пользователей и отслеживание изменения статуса всех пользователей
* Администратор 3 – организация размещения данных, назначение прав, планирование резервного копирования, хранение резервных копий и совместно с Адм1 восстановление данных
* Администратор безопасности – участвует в разработки матрицы доступа, контролирует соблюдение политики безопасности ИС при ее эксплуатации, отслеживает информацию о БС, тестирует ИС на предмет брешей в защите.
* Аудитор – настройка подсистемы регистрации, организация архивирования и хранения журналов регистрации и их анализ

**Состав пакета нормативных документов**:

1. Положение о локальной сети компании.
2. Инструкция администратору серверов.
3. Инструкция администратору бд.
4. Инструкция пользователю.
5. Инструкция администратору информационной безопасности.
6. Инструкция аудитору.
7. Процедура оформления доступа к ресурсам.
8. Инструкция по резервному копированию и восстановлению информации.
9. Инструкция по антивирусной безопасности.
10. Инструкция о парольной защите.

**3. Системное администрирование. Сетевое администрирование.**

Системное администрирование – процесс управления, целью которого является приведение сети в соответствие целям и задачам, для которых эта сеть предназначена.

Сетевое администрирование – действия, направленные на поддержку и настройку работоспособности корпоративной сети и систем на корпоративных машинах, решение нового спектра задач управления ИС, включающих в себя учет распределенных ресурсов, электронное распространение ПО, контроль лицензий, анализ трафика, управление пропускной способностью сети, перераспределение серверной нагрузки, отслеживание состояния отдельных настольных систем

**4. Задачи администрирования ИС. Основные проблемы, с которыми сталкивается администратор сети.**

**Задачи администрирования ИС:**

1. Установка и настройка сети.
2. Поддержка ее работоспособности
3. Установка базового ПО
4. Мониторинг сети

**Основные проблемы администрирования информационной сети:**

Проблема - При всех своих достоинствах сети TCP/IP имеют один врожденный недостаток - отсутствие встроенных способов защиты информации от несанкционированного доступа.

**5. Идентификация. Аутентификация. Авторизация. Пример процесса идентификации, аутентификации и авторизации.**

Идентификация – процесс опознания определенного объекта по его свойствам. (Опознание аккаунта в системе по имени пользователя)

Аутентификация – процесс подтверждения подлинности объекта, с помощью какого-либо уникального значения или свойства объекта (Вход пользователя в операционную систему Windows по паролю)

Авторизация – процесс проверки прав на доступ и предоставление доступа после того, как объект был идентифицирован или аутентифицирован, он может запросить доступ к каким-либо ресурсам.

**6. Технологии идентификации.**

**Идентификация –** процедура распознавания субъекта по его уникальному идентификатору, присвоенному данному субъекту ранее и занесенному в БД в момент его регистрации в качестве легального пользователя системы (процесс опознания, определения субъекта по его свойствам).

Идентификатором может быть: номер телефона, номер паспорта, e-mail, номер страницы в социальной сети и т.д.

Существует штрих-кодная идентификация, радиочастотная, биометрическая и идентификация на основе карт с магнитной полосой.

Штрих-кодная используется производителями товаров для автоматизации товаропродвижения. Достоинства: снижение бумажного документооборота и количества ошибок при вводе информации, повышение скорости обслуживания клиентов, автоматизации основных технологических процессов товаропродвижения на всех этапах от производителя до конечного покупателя. Недостатки: данные штрих-кода не могут дополняться, небольшой объем данных (не больше 50 байт), медленное занесение данных на метку (печатается символ, наклеивание липкой этикетки происходит вручную), данные на метке представлены в открытом виде и не защищают от подделок и краж, недолговечны (не защищена от пыли, сырости, грязи).

Радиочастотная (RFID), в ней развиты почти все недостатки штрих-кодовой. Внедряются во многие отрасли мирового хозяйства.

Позволяют получать информацию о предмете без прямого контакта. Дистанция для считывания и записи информации от нескольких миллиметров до нескольких метров в зависимости от используемых технологий (от частоты в пределах 125кГц-5.8ГГц). Большинство применяется для идентификации сотрудников корпораций смарт-карт с применением компонент производства Ангстрем, HID, Atmel, Mifare, EM Microelectronic Marin, Microchip и др. чаще всего используют несущие частоты 125 кГц или 13,56МГц.

Биометрическая основана на применении статистического анализа биологических наблюдений и явлений. Биометрическая характеристика – это измеримая физиологическая или поведенческая черта человека.

Биометрические системы отличаются, в основном, объектами и способами измерений. Пользователь посредством регистрирующего устройства (сканера или камеры) предоставляет системе образец – опознаваемое, необработанное изображение или запись физиологической, или поведенческой характеристики.

Биометрический образец обрабатывается для получения информации об отличительных признаках, в результате чего получается эталонный идентификатор пользователя или эталон для проверки, который представляет собой числовую последовательность, при этом сам образец невозможно восстановить из эталона.

Снятая в процессе идентификации характеристика сравнивается с ЭИП. Поскольку эти два значения (полученное при попытке доступа и ЭИП) полностью никогда не совпадают, то для принятия положительного решения о доступе степень совпадения должна превышать определенную настраиваемую пороговую величину.

При этом эффективность биометрических систем характеризуется коэффициентом ошибочных отказов и коэффициентом ошибочных подтверждений.

Карты с магнитной полосой срабатывают при проведении в определенном направлении с определенной скоростью по щели считывателя. Современные магнитные полосы изготовлены из материалов, требующих сильных магнитных полей для записи и уничтожения информации, с целью сохранности информации от случайного размагничивания.

Преимущество – низкая стоимость. Недостатки: ограничение объема информации, незащищенность от копирования, чувствительность к загрязнению, к царапинам, изломам, влаги, короткий срок службы (не более 3 лет).

**7.Технологии аутентификации. Протоколы аутентификации. Использование цифрового сертификата, смарт-карт, usb-ключей.**

**Аутентификация –** процедура проверки подлинности входящего в систему объекта, предъявившего свой идентификатор (подтверждение подлинности с помощью какого-либо уникального значения/свойства объекта).

Пример: sim-карта и телефон, когда вставляете новую sim-карту, требуют ввести пин-код, после успешного ввода телефон начинает работать. То есть, двухфакторная аутентификация. Надо иметь персональный носитель (SIM-карту) и знать личный PIN-код.

**Протокол аутентификации** должен выполнять по крайней мере две задачи:

1. Безопасно передавать транзакции от запросчика в БД аутентификации и на любой другой компьютер, на котором размещен соответствующий ресурс.
2. Безопасно и надежно хранить пароль или маркер.

Протокол аутентификации должен защитить введенную пользователем информацию при пересылке в бд аутентификации (т.е. SAM, Security Account Manager или AD). Для этого протокол подписывает, скрывает или шифрует транзакцию. Кроме того, ей присваивается временная метка, чтобы взломщик не мог воспользоваться учетными данными в будущем. Чтобы не позволить немедленно извлечь пароль пользователя из бд, протокол должен обеспечить скрытное хранение паролей в бд аутентификации.

В течение более чем десяти лет протоколы аутентификации в основном обеспечивали защиту путем сохранения паролей в скрытой форме (обычно хешированной) в бд аутентификации и полного запрета на передачу паролей между запросчиком и бд аутентификации простым текстом (даже в скрытой форме).

В протоколах аутентификации используется процесс запрос-ответ, поэтому пароль никогда не передается через сеть.

Процесс запрос-ответ выглядит следующим образом:

1. Компьютер получает данные для идентификации и аутентификации от пользователя и запрашивает аутентификацию на соответствующем сервере.
2. Сервер аутентификации генерирует случайное произвольное значение (называемое запросом - challenge) и посылает его запросчику.
3. Запросчик получает запрос и производит над ним и скрытой формой пароля математические операции, а затем передает результат (называемый ответом - response) серверу аутентификации.
4. Сервер аутентификации также выполняет математические манипуляции с запросом методом, идентичным используемому на рабочей станции, и сравнивает результат с полученным ответом. Если результаты совпадают, то запросчик считается успешно аутентифицированным.

Часть протоколов сетевой аутентификации были разработаны специально для обеспечения удаленного доступа к информационным ресурсам посредством открытых каналов связи (например, телефонные лини, internet). Например, такие протоколы: PAP, CHAP, EAP, RADIUS, TACACS и др.

Протокол аутентификации RADIUS рассматривается как механизм аутентификации и авторизации удалённых пользователей в условиях распределённой сетевой инфраструктуры, предоставляющий централизованные услуги по проверке подлинности и учёту для служб удалённого доступа.

В рамках стандарта выделяются следующие роли:

* Клиент RADIUS принимает от пользователей запросы на аутентификацию. Все принятые запросы переадресовываются серверу RADIUS для последующей аутентификации и авторизации. Как правило, в качестве клиента протокола RADIUS выступает сервер удалённого доступа.
* Сервер RADIUS. Основная задача заключается в централизованной обработке информации, предоставленной клиентами RADIUS. Один сервер способен обслуживать несколько клиентов RADIUS. Сервер осуществляет проверку подлинности пользователя и его полномочий. При этом в зависимости от реализации сервера RADIUS для проверки подлинности используются различные базы данных учётных записей.
* Посредник RADIUS. Взаимодействие клиентов и серверов RADIUS осуществляется посредством специальных сообщений. В распределённых сетях клиент и сервер RADIUS могут быть разделены различными сетевыми устройствами (такими, например, как маршрутизатор). Под посредником RADIUS понимается сетевое устройство, способное осуществлять перенаправление сообщений протокола RADIUS.

Поддержка протокола RADIUS реализована на многих современных платформах, что позволяет использовать его в межплатформенных решениях.

В качестве примера сервера и посредника RADIUS можно привести реализованную в Windows Server 2003 службу проверки подлинности в Интернете (Internet Authentication Service, IAS). Эта служба позиционируется как механизм централизованной аутентификации и авторизации пользователей, использующих различные способы подключений к сети. Служба IAS интегрирована с другими сетевыми службами Windows Server 2003, такими, как служба маршрутизации и удалённого доступа и служба каталога Active Directory.

**Использование цифрового сертификата**

Механизмы аутентификации с применением сертификатов обычно используют протокол запрос-ответ. Согласно этому протоколу, сервер аутентификации направляет пользователю последовательность символов, называемую запросом, а ПО клиентского компьютера для генерирования ответа вырабатывает с помощью закрытого ключа пользователя цифровую подпись под запросом от сервера аутентификации.

Общий процесс подтверждения подлинности пользователя состоит из следующих стадий:

* получение открытого ключа СА (одноразовый процесс),
* получение по некоторому незащищенному каналу от этого пользователя его сертификата открытого ключа.

Аутентификация с открытым ключом используется как защищенный механизм аутентификации в таких протоколах как SSL, а также может использоваться как один из методов аутентификации в рамках рассмотренных протоколов Kerberos и RADIUS.

SSL (протокол защищенных сокетов) – криптографический протокол, призванный обеспечить безопасную передачу данных по сети Интернет.

**Использование смарт-карт и USB-ключей**

Несмотря на то, что криптография с открытым ключом согласно спецификации, Х.509 может обеспечивать строгую аутентификацию пользователя, сам по себе незащищенный закрытый ключ подобен паспорту без фотографии.

Закрытый ключ, хранящийся на жёстком диске компьютера владельца, уязвим по отношению к прямым и сетевым атакам. Достаточно подготовленный злоумышленник может похитить персональный ключ пользователя и с помощью этого ключа представляться этим пользователем.

Защита ключа с помощью пароля помогает, но недостаточно эффективно – пароли уязвимы по отношению ко многим атакам. Требуется более безопасное хранилище.

**Смарт-карты –** пластиковые карты стандартного размера банковской карты, имеющие встроенную микросхему. Они находят всё более широкое применение в различных областях, от систем накопительных скидок до кредитных и дебетовых карт, студенческих билетов и телефонов стандарта GSM.

Для использования смарт-карт в компьютерных системах необходимо устройство чтения смарт-карт. Устройство чтения (считыватель), **–** большинство подобных оконечных устройств, или устройств сопряжения (IFD – Interface Device), способны как считывать, так и записывать информацию, если позволяют возможности смарт-карты и права доступа.

Устройства чтения смарт-карт могут подключаться к компьютеру посредством последовательного порта, слота PCMCIA или USB. Устройство чтения смарт-карт также может быть встроено в клавиатуру. Как правило, для доступа к защищенной информации, хранящейся в памяти смарт-карты, требуется пароль, называемый PIN-кодом.

**USB-ключи** достаточно привлекательны, поскольку USB стал стандартным портом для подключения периферийных устройств и организации не нужно приобретать для пользователей какие бы то ни было считыватели.

Аутентификацию на основе смарт-карт и USB-ключей сложнее всего обойти, так как используется уникальный физический объект, которым должен обладать человек, чтобы войти в систему. В отличие от паролей, владелец быстро узнаёт о краже и может сразу принять необходимые меры для предотвращения её негативных последствий. Кроме того, реализуется двухфакторная аутентификация.

Микропроцессорные смарт-карты и USB-ключи могут повысить надёжность служб PKI: смарт-карта может использоваться для безопасного хранения закрытых ключей пользователя, а также для безопасного выполнения криптографических преобразований. Безусловно, данные устройства аутентификации не обеспечивают абсолютную безопасность, но надёжность их защиты намного превосходит возможности обычного настольного компьютера.

Для хранения и использования закрытого ключа разработчики используют различные подходы.

1. Наиболее простой – использование устройства аутентификации в качестве защищенного носителя аутентификационной информации: при необходимости карта экспортирует закрытый ключ, и криптографические операции осуществляются на рабочей станции. Этот подход является не самым совершенным с точки зрения безопасности, зато относительно легко реализуемым и предъявляющим невысокие требования к устройству аутентификации.

Два других подхода более безопасны, поскольку предполагают выполнение устройством аутентификации криптографические операций.

1. Пользователь генерирует ключи на рабочей станции и сохраняет их в памяти устройства.
2. Пользователь генерирует ключи при помощи устройства.

В обоих случаях, после того как закрытый ключ сохранён, его нельзя извлечь из устройства и получить любым другим способом.

**8. Организация администрирования ИС. Управление проектами в ИС. Управление конфигурацией ИС. Аудит ИС. Тестирование ИС.**

**Организация администрирования ИС**

Администрирование связано также с планированием размещения оборудования (компьютеров) и созданием необходимого микроклимата в помещениях с компьютерами, особенно в серверной. В серверных рекомендуется создавать специальную централизованную систему кондиционирования воздуха, располагая разнесённые сервера в одном месте, так как каждое превышение температуры на 8 выше стандартных 21 на 50% уменьшает надёжность работы компьютеров.

Так как ИС является динамическим информационным отображением предметной области, то желательно, чтобы администратор своевременно получал информацию о перспективах развития объекта, для которого создаётся информационная система. Руководство организации и администратор согласовывают:

* цели и направления,
* сроки создания ИС и её развития,
* очередность подключения пользователей.

На администраторов компьютерных систем ложится задача обслуживания ИС и сохранения настроек ПО на рабочих машинах.

Процесс администрирования должен осуществляться непосредственно со специальных административных станций, выделенных в логический (или физический) сегмент. В зависимости от потенциальных возможностей администратора и ценности циркулирующей в системе информации, станции администрирования можно размещать в защищённой зоне и снабжать дополнительными средствами защиты.

Вполне естественным является желание администратора избавить себя от лишних забот, проведя максимальную автоматизацию рутинных процедур. Многие из них разрабатывают с этой целью небольшие утилиты и скрипты. На рынке представлено большое количество специализированных средств, позволяющих перевести труд администраторов на качественно новый уровень и унифицировать управление разнородными системами.

Поэтому есть принципы, соблюдение которых позволяет сделать службу администрирования экономически более выгодной по сравнению с традиционными подходами:

1) централизованное администрирование, поддержанное системой репликаций (экономия на квалифицированном персонале, меньше потерь из-за ошибок управления);

2) возможность выполнения основных объемов работ по администрированию удаленным образом (экономия на транспортных и командировочных расходах);

3) адаптивное управление маршрутом трафика данных в зависимости от условий и качества каналов связи (экономия на междугороднем трафике и тарифных планах провайдеров).

Основной функцией **управления проектами** является планирование выполнения необходимых для реализации управления работ (создание рабочего и иных планов работ).

Рабочий план должен соответствовать общим требованиям, согласно которым в нем должны присутствовать следующие компоненты:

* указание цели;
* определение требуемых задач;
* распределение обязанностей;
* указание сроков, а также:
* требуемые ресурсы;
* источники материально-технической поддержки;
* бюджетные ассигнования;
* определение количественного аспекта;
* сочетание с другими компонентами;
* промежуточная отчётность и др.

Обычно выделяют восемь этапов процесса планирования.

*Этап 1*. Целеполагание (формулировка целей). Он нацелен на выявление результатов, которые хотелось бы получить (достичь)? Это – самый трудный и не формализуемый этап.

*Этап 2*. Подбор способов достижения поставленных целей, их анализ и оценка. Обычно существует несколько способов достижения цели. Важно не только их выявить, но и определить наилучший из них.

*Этап 3*. Составление перечня необходимых действий. Определение конкретных действий, необходимых для осуществления (реализации) выбранного на предыдущем этапе варианта достижения поставленных целей.

*Этап 4*. Составление программы работ (плана мероприятий). На этом этапе устанавливают порядок выполнения намеченных на предыдущем этапе действий с учётом того обстоятельства, что многие из них связаны между собой.

*Этап 5*. Анализ ресурсов. В процессе его выполнения определяют необходимые для реализации плана выполнения намеченных работ интервалы времени, материальные (в том числе программно-технические), финансовые, информационные и кадровые ресурсы.

*Этап 6*. Анализ разработанного варианта плана. Здесь следует установить: решает ли разработанный план цели, поставленные на первом этапе; являются ли приемлемыми затраты ресурсов; есть ли соображения по улучшению плана, возникшие в ходе его разработки при движении от второго к пятому этапам. В результате проведения этой работы может оказаться целесообразным вернуться к этапам 2, 3, и даже к первому этапу.

*Этап 7*. Подготовка детального плана действий. На этом этапе необходимо детализировать разработанный на предыдущих этапах план, выбрать согласованные сроки выполнения отдельных видов работ, рассчитать необходимые ресурсы, установить ответственных за отдельные участки работы.

*Этап 8*. Контроль выполнения плана, внесение необходимых изменений (в случае необходимости).

**Управление конфигурацией ИС –** один из вспомогательных процессов, поддерживающих основные процессы жизненного цикла ПО, прежде всего процессы разработки и сопровождения ПО ИС.

При создании проектов сложных ИС, состоящих из многих компонентов, каждый из которых может иметь разновидности или версии, возникает проблема учёта их связей и функций, создания унифицированной структуры и обеспечения развития всей системы.

Управление конфигурацией позволяет организовать, систематически учитывать и контролировать внесение изменений в ПО на всех стадиях ЖЦ. Общие принципы и рекомендации конфигурационного учёта, планирования и управления конфигурациями ПО отражены в проекте стандарта ISO/IEC 12207.

Каждый процесс характеризуется определёнными задачами и методами их решения, исходными данными, полученными на предыдущем этапе, и результатами. Результатами анализа, в частности, являются функциональные модели, информационные модели и соответствующие им диаграммы.

ЖЦ ПО носит итерационный характер: результаты очередного этапа часто вызывают изменения в проектных решениях, выработанных на более ранних этапах.

К управлению конфигурацией следует отнести функции анализа производительности и оптимизации системы.

Управляемость сети – возможность централизованно контролировать состояние основных элементов сети, выявлять и разрешать проблемы, возникающие при работе сети, выполнять анализ производительности и планировать развитие сети.

Совместимость – способность сети включать в себя самое разнообразное программное и аппаратное обеспечение, то есть в ней могут сосуществовать различные ОС, поддерживающие разные стеки коммуникационных протоколов.

Модульность – одно из неотъемлемых и естественных свойств вычислительных сетей, отражается в многоуровневом представлении коммуникационных протоколов в конечных узлах сети.

Расширяемость – возможность легкого добавления отдельных элементов сети (пользователей, компьютеров, приложений, служб), наращивания длины сегментов сети и замены существующей аппаратуры более мощной.

Масштабируемость – способность сети плавно увеличивать вычислительную мощность, сеть позволяет наращивать количество узлов и протяженность связей в очень широких пределах, при этом производительность сети не ухудшается.

Тиражирование (репликация) – создание дубликатов данных.

Оценка состояния, функциональности, возможностей ИС осуществляется путем ее аудита и тестирования.

**Аудит ИС** – это проверка и оценка практики использования ИТ-систем в организации, осуществляемая специализированной независимой организацией.

Выделяют две причины необходимости проведения такого аудита:

1. Информационным технологиям большинства Российских компаний свойственен эволюционный путь создания и дальнейшего их развития. Он характеризуется тем, что информационные системы включаются в инфраструктуру ИТ или модернизируются по мере возникновения необходимости и (или) по мере возможности (в том числе и финансовой). В итоге в ИТ-инфраструктуре формируется сложная (порой разнородная) и поэтому плохо управляемая совокупность программно-технических и системных платформ. Даже если ИС прошла все стадии создания, последующие изменения бизнес-процессов или введение новых приложений могут привести к тому, что параметры её программно-аппаратных платформ перестанут соответствовать требованиям бизнеса.
2. Вторая причина связана с зависимостью успешности бизнеса от способности управленцев вовремя получать и быстро обрабатывать нужную информацию. Очевидно, что современный руководитель любого звена не в состоянии одновременно быть компетентным в различных областях, обеспечивать получение и обработку разноплановой информации.

Система аудита ИС может быть разделена на три составные части:

* внутренний контроль, осуществляемый организацией, эксплуатирующей ИС;
* внешний контроль, осуществляемый внешней организацией в рамках разделения полномочий организации, эксплуатирующей ИС, и вышестоящего ведомства или организации;
* независимый информационный аудит.

При этом в общем случае выделяют следующие виды аудита ИТ – это аудит:

* процессов управления службой ИТ,
* структуры службы ИТ,
* информационной системы,
* ТЗ и проектной документации на создание ИС,
* систем резервирования данных.

Результаты аудита используются почти всеми задачами администрирования ИС.

**С целью обеспечения непротиворечивости получаемой информации доступ к подсистеме аудита должен быть ограничен. Причём лицо, ответственное за подсистему аудита не должно иметь административных полномочий по управлению системой и данным, по которым аудит ведётся. Для определения лиц и организаций, имеющих право осуществлять аудит ИС, рекомендуется разработать и принять обязательные требования к информационным аудиторам и определить случаи их применения.**

Считается, что проведение информационного аудита должно стать необходимым этапом получения бюджетных средств для создания или развития ИТ-систем.

**Тестирование ИС –** процесс исполнения программы с целью обнаружения ошибок.

Регрессионное тестирование – это тестирование, проводимое после усовершенствования функций программы или внесения в неё изменений. Одно из средств тестирования QA (ныне – Quality Works) представляет интегрированную, многоплатформенную среду разработки автоматизированных тестов любого уровня, включая тесты регрессии для приложений с графическим интерфейсом пользователя.

Критерии тестирования включают:

* описание тестов;
* фиксацию повторения действий оператора (возможность фиксировать данные, вводимые оператором с помощью клавиатуры, мыши и т.д., редактировать их и воспроизводить в тестовых примерах);
* автоматический запуск тестовых примеров;
* регрессионное тестирование (возможность повторения и модификации ранее выполненных тестов для определения различий в системе и/или среде);
* автоматизированный анализ результатов тестирования и исключительных ситуаций в процессе тестирования, включая сравнение ожидаемых и реальных результатов, сравнение файлов, статистический анализ результатов; обращения к операторам, процедурам и переменным; защиту от несанкционированного доступа и др.;
* анализ производительности. Анализируемые параметры производительности могут включать использование центрального процессора, памяти, обращения к определённым элементам данных и (или) сегментам кода, временные характеристики и т.д.

**9. Эксплуатация и сопровождение ИС. Управление рисками и инцидентами. Оценка рисков.**

Эксплуатация ИС – направление, целью которого является минимизация вероятности возникновения проблем качества работы ИС на платформе предприятия.

Эксплуатация включает работы по внедрению компонентов ПО в эксплуатацию, в том числе:

* конфигурирование БД и рабочих мест пользователей,
* обеспечение эксплуатационной документацией,
* проведение обучения персонала,
* непосредственно эксплуатацию,
* локализацию проблем и устранение причин их возникновения,
* модификацию ПО в рамках установленного регламента,
* подготовку предложений по совершенствованию, развитию и модернизации системы.

Ожидаемые результаты должны рассматриваться с учётом вероятной отсрочки в улучшении проектных и эксплуатационных характеристик.

Техническое обслуживание и модернизация.

Техническое обслуживание заключается в очистки от пыли, смазки вентиляторов, подтяжки креплений, контроле состояния аккумуляторов, изменении физической топологии сети и т. п., осуществляется службой технической поддержки. Грамотное формулирование заявок на изменение аппаратной конфигурации, организация закупки дополнительных лицензий или обновленной версии ПО – задача администратора.

Важным вопросом сопровождения ИС является мониторинг работы сетевого и иного вычислительного оборудования. Задачу оперативного управления ИС выполняет администратор системы. Принято обращать внимание на критически важные инциденты. Затем рекомендуется осуществлять контроль сроков исполнения, оптимизировать контролируемы параметры и др.

**Управление рисками и инцидентами.**

Определение рисков – сложная задача, их успешное выявление и ликвидация зависят от умения их распознавать.

Выделяют восемь наиболее опасных рисков:

1) недостаточное внимание к проекту со стороны руководства заказчика (компании) и недостаточное в нём участие;

2) неконкретная постановка задачи или непонимание сторонами конечных целей проекта;

3) изменения, вносимые заказчиком в процессе реализации проекта;

4) недостаточная квалификация работников;

5) отсутствия мотивации сотрудников заказчика, противодействие персонала;

6) срыв сроков;

7) технические проблемы;

8) недостаточное или нестабильное финансирование.

Недостаточное или нестабильное финансирование является одним из типовых рисков, независящих как от разработчиков проектов, так и специалистов, эксплуатирующих ИС. Предлагается определять “точку невозврата”, то есть ситуацию, при которой теряется весь бюджет. Оценивая риск прекращения финансирования, важно определить его появление на соответствующих этапах.

К рискам администрирования ИС относят:

1. сокращение установленных в соответствующих планах (графиках)
2. сроков выполнения работ;
3. увеличение стоимости сопровождения, эксплуатации и
4. администрирования ИС из-за системных и иных ошибок,
5. недостаточного уровня поддержки со стороны руководства и
6. администраторов ИС;
7. сложность эксплуатации системы;
8. несоблюдение условий безопасности ИС и хранящихся в ней данных;
9. сбои;
10. увольнение администраторов и специалистов, осуществляющих
11. эксплуатацию и поддержку ИС и др.

**Оценка рисков:**

Количественную оценкубольшинства организационно-управленческих рисков для ИТ-проектов (ИС) оценить трудно, так же, как и его экономическую эффективность, так как они обычно носят характер качественного улучшения и выполняются в условиях относительно низкой неопределённости. При этом обычно все наиболее важные риски случаются до начала выполнения проекта.

Качественную оценку рисков в виде низкий, средний, высокий, некоторые специалисты считают возможным перевести в качественно-количественные показатели. При этом они определяют высокую вероятность появления риска в 70% и выше (или 3 балла), среднюю – от 40 до 70% (2 балла) и низкую – менее 40% (1 балл). Другой критерий предполагает, что если воздействие приведёт к потере менее 1% бюджета или затянет сроки исполнения проекта менее чем на 5% отведённого времени, то его можно считать низким. Соответственно при 1–5% бюджета и 5– 10% дополнительного времени – средним. Если бюджет превышен больше чем на 5%, а сроки – более чем на 10%, то высоким.

Проведение внешней экспертизы по оценке рисков способствует сокращению серьёзных системных ошибок с далеко идущими последствиями, но может привести, как утверждают некоторые специалисты, к погрешности расчётов.

Для определения рисков некоторые специалисты рекомендуют составлять таблицу с полями или столбцами, позволяющими получать ответы на следующие вопросы:

* *что может случиться;*
* *какова вероятность что это случится; на каком этапе проекта это может произойти;*
* *сколь сильно это может повлиять на результаты проекта;*
* *что необходимо делать, чтобы это не случилось;*
* *что делать, если это все-таки произойдёт?*

**10. Управление безопасностью. Виды функциональной безопасности ИС. Проверка полномочий, проверка подлинности. Фундаментальные принципы обеспечения безопасности. Многоуровневая безопасность. Классы и уровни допуска или доступа к информации. Методы обеспечения сохранности информации.**

Важным компонентом администрирования системы является обеспечение информационной безопасности: составление плана доступа пользователей к ресурсам (в соответствии с принятой в компании политикой информационной безопасности) и контроль его исполнения.

К функциям обеспечения безопасности относятся также отслеживание появления различных уязвимостей в используемых ОС, организация получения и установки “заплаток” (patches).

Функционально безопасность ИС можно разделить на технологическую, логическую и физическую безопасности.

• Для обеспечения **технологической безопасности** в ИС используют “зеркальные” серверы, двойные жёсткие диски, надёжные системы бесперебойного питания и др.

• **Логическая безопасность** заключается в использовании программных средств борьбы с компьютерными вирусами, защиты от несанкционированного доступа, идентификации и кодирования информации.

• **Физическая безопасность** включает персонал, меры и преграды, препятствующие проникновению несанкционированных лиц на недоступные для них объекты.

Характеристики надежности:

* Аспектом общей надёжности ИС является её безопасность (security), то есть способность системы защитить данные от НСД. В распределённой системе это сделать сложнее, чем в централизованной. В сетях сообщения передаются по линиям связи, часто проходящим через общедоступные помещения, в которых могут быть установлены средства прослушивания линий. Другим уязвимым местом могут быть оставленные без присмотра ПК. Также, имеется потенциальная угроза взлома защиты сети от неавторизованных пользователей, если сеть имеет выходы в глобальные сети общего пользования.
* Отказоустойчивость (fault tolerance) – способность системы скрыть от пользователя отказ отдельных её элементов. Например, если копии таблицы бд хранятся одновременно на нескольких файловых серверах, то пользователи могут просто не заметить отказ одного из них. В отказоустойчивой системе отказ одного из её элементов приводит к некоторому снижению качества её работы (деградации), а не к полному останову.

Для защиты информации в информационных компьютерных сетях используют специальные программные, технические и программно-технические средства.

С целью защиты сетей и контроля доступа в них используют:

– фильтры пакетов, запрещающие установление соединений, пересекающих границы защищаемой сети;

– фильтрующие маршрутизаторы, реализующие алгоритмы анализа адресов отправления и назначения пакетов в сети;

– шлюзы прикладных программ, проверяющие права доступа к программам.

Необходимо поддерживать два **фундаментальных принципа**: проверку полномочий (санкционирование) и проверку подлинности (аутентификация).

**Проверка полномочий** основана на том, что для каждого пользователя или процесса ИС устанавливаться набор санкционированных действий, которые он может выполнять по отношению к определенным объектам. Проверка полномочий сама по себе недостаточна для обеспечения даже минимального уровня безопасности. Если, например, процесс «2» сможет успешно выдать себя за процесс «1», то он сможет выполнять действия и операции, доступные только процессу «1». Поэтому необходимы дополнительные меры.

**Проверка подлинности** подтверждает, что пользователь или процесс, пытающийся выполнить санкционированные действия, действительно является тем, за кого он себя выдает. В безопасной среде должна поддерживаться проверка подлинности, способная обеспечить надёжную верификацию идентификаторов, предъявляемых пользователями или процессами.

Сочетание средств проверки полномочий и проверки подлинности является мощным оружием в борьбе за безопасность ИС.

Но модель безопасности, основанная на базовых механизмах проверки полномочий и подлинности, не решает проблем, связанных с хищениями пользовательских идентификаторов и паролей или злонамеренными действиями пользователей, обладающих полномочиями, например, администраторов ИС.

В системе может храниться информация, относящаяся к разным классам безопасности (от полностью открытой до совершенно секретной), поэтому любые пользователи системы не могут иметь разрешение на доступ к самой секретной информации. Для этого существует организованная среда с многоуровневой безопасностью.

**Многоуровневая безопасность** означает, что в системе хранится информация, относящаяся к разным классам безопасности и часть пользователей, не имеет доступа к информации, относящейся к высшему классу безопасности. Субъект имеет доступ к объекту, если уровень его допуска такой же или ниже, чем класс объекта. При этом пользователь, имеющий низший уровень допуска, должен иметь возможность выполнять свою работу в системе, содержащей в бд совершенно секретные данные, но не должен иметь доступа к ним. Информация и данные подвергаются классификации, а каждый субъект получает определённый уровень допуска к соответствующим классам данных (объектов).

**Классы и уровни допуска или доступа к информации**

В военных и государственных ведомствах применяют следующую иерархию классов (сверху вниз):

• совершенно секретно;

• секретно;

• конфиденциально;

• без грифа секретности.

В частных компаниях возможны уровни иерархии (сверху вниз):

• секретно;

• для ограниченного распространения;

• конфиденциально;

• для служебного пользования;

• для неограниченного распространения.

Многоэкземплярность – общепринятая модель реализации бд с многоуровневой безопасностью, заключается в том, что в рамках одного отношения может существовать множество кортежей (записей) с одним и тем же значением первичного ключа, например, данные с различными классами доступа могут быть реализованы в виде нескольких бд.

Защита носителей информации

Одной из важных проблем информационной безопасности является организация защиты электронных данных, хранящихся на различных носителях.

Защита данных от НСД предполагает:

1. Обеспечение парольного входа в систему: регистрация пользователей, назначение и изменение паролей.

2. Обеспечение защиты конкретных данных: определение прав доступа групп пользователей и отдельных пользователей, определение допустимых операций над данными для отдельных пользователей, выбор/создание программно-технологических средств защиты данных; шифрование информации с целью защиты данных от несанкционированного использования.

3. Тестирование средств защиты данных.

4. Фиксация попыток несанкционированного доступа к информации.

5. Исследование возникающих случаев нарушения защиты данных и проведение мероприятий по их предотвращению.

Обеспечение сохранности информации производится путём применения специальных мер организации хранения, восстановления (регенерации) информации, специальных устройств резервирования.

Качество обеспечения сохранности информации зависит от её целостности (точности, полноты) и готовности к постоянному использованию.

Важное значение для информации имеют методы её хранения и сохранения. Специалисты предлагают **методы обеспечения сохранности электронных** данных вообще и в Интернете в частности:

• постоянная миграция материала к наиболее современным аппаратурно-программным средствам (т.е. непрерывная перезапись ресурса);

• сохранение исходного формата и средств раскрытия содержания материала;

• копирование (архивирование);

• защита от несанкционированного использования, замены, искажения и удаления;

• защита от компьютерных вирусов и неполадок в электрических и компьютерных сетях.

К сфере защиты данных относятся также их сохранность и восстановление после сбоя системы.

**11. Резервное копирование и восстановление данных. Виды резервного копирования.**

Важной особенностью систем хранения данных является непрерывная их защита (Continuous Data Protection, CDP). Для защиты данных, хранящихся на файл-сервере, применяют **резервное копирование**, ленточный автозагрузчик и др. Открытые файлы не копируются на ленту. Операция резервного копирования создаёт дополнительную нагрузку на сервер. В этом случае резервное копирование можно осуществлять в нерабочее время (ночью, в выходные и праздничные дни и др.), то есть в специально выделяемый для этого период времени – окно резервного копирования.

С целью обеспечения достоверности и постоянной работоспособности ИС периодически вручную или автоматически осуществляется копирование ИС.

Причины копирования (различные структурные изменения в ИС):

• создание или удаление табличного пространства (содержат объекты бд);

• добавление или переименование (перемещение) файла данных в существующем табличном пространстве;

• добавление, переименование (перемещение) или удаление журнала повторения и др.

При этом резервное копирование может осуществляться непосредственно перед изменениями и после них.

**Виды резервного копирования:**

1. Непротиворечивое (холодное) резервное копирование, когда копии создаются, в случае закрытой для пользователей ИС. Копия ИС, созданной в автономном режиме, содержит: все файлы данных, журналы повторов и управляющие файлы. После остановки ИС, все её файлы копируются на один из “backup” дисков. По окончании копирования осуществляется перезагрузка ИС.
2. Резервное (горячее) копирование в оперативном режиме, к примеру, когда ИС всё время находится в оперативном режиме и доступна пользователям. Применяют, когда работают два сервера: основной, обрабатывающий запросы пользователей, и резервный, который продолжает работу основного сервера в случае его отказа. Состояние хранилищ данных на основном и резервном серверах согласовано и поддерживается ИС автоматически. Но не избавляет от необходимости хранения резервных копий данных.

Достаточно обеспечить наличие образа ИС (архив всех файлов, а также управляющих файлов – снимок ИС на определенный момент времени; проще всего такой снимок получить, остановив ИС и сделав резервную копию всех указанных файлов).

**12. Концепция управления ИТ-службами. ITSM. Преимущества внедрения концепции ITSM.**

**ITSM –** концепция управления ИТ-службами, которая предлагает новый взгляд на организацию функционирования ИТ-подразделений, порядок управления этими подразделениями, пути повышения эффективности использования ресурсов. ITSM возникла в результате принципиального изменения сегодняшней роли ИТ-подразделений. Бизнес-процессы компаний тесно увязаны с приложениями, техническими ресурсами и деятельностью персонала отделов автоматизации, поэтому их эффективность – один из решающих факторов эффективности компании в целом.

Идея внедрения ITSM состоит в том, чтобы ИТ-отдел перестал быть вспомогательным элементом для основного бизнеса компании, ответственным только за работу отдельных серверов, сетей и приложений, применяющихся в компании. Отдел автоматизации становится полноправным участником бизнеса, выступая в роли поставщика определенных услуг для бизнес-подразделений, а отношения между ними формализуются как отношения «поставщик услуг – потребитель услуг».

Типичные примеры ИТ-процессов:

• установка нового ПО,

• ликвидация проблем в сети,

• процесс перехода на новую резервную систему и т.д.

Важнейшая составляющая реализации ITSM – разработка формализованных процессов ИТ-отдела:

Для каждого процесса определяется:

• последовательность выполнения работ,

• необходимые ресурсы,

• затраты времени,

• средства автоматизации,

• контроль качества.

**Преимущества внедрения:**

1. Улучшение качества обслуживания клиентов организации.

2. Повышение качества взаимодействия различных бизнес-подразделений организации.

3. Улучшение качества обслуживания потребителей ИТ-услуг внутри организации.

4. Контроль за текущим состоянием ИТ-инфраструктуры в целом, отдельных сервисов и компонент.

5. Доступность в реальном времени и в удобном для понимания виде информации о:

– работоспособности конкретных приложений

– загруженности каждого из критичных ресурсов

– соответствии предоставляемых сервисов требуемому уровню позволит реально оценивать текущее состояние дел и при необходимости оперативно реагировать на критичные состояния.

6. Анализ эффективности функционирования ИТ-инфраструктуры и ее элементов. За счет предоставления возможности обработки и отображения интегральных характеристик уровня использования существующих ресурсов:

– загруженности серверов

– загруженности каналов связи

– качества работы приложений позволит оценить эффективность использования ресурсов и планировать дальнейшее развитие ИТ - инфраструктуры.

7. Повышение качества взаимодействия различных ИТ-подразделений между собой, путем повышения прозрачности происходящих в ИТ-инфраструктуре процессов. Взаимоотношения между подразделениями ИТ, состояние происходящих там процессов также, как и процессов, ориентированных на внешних пользователей, могут быть отражены при помощи различных метрик и сделаны доступными для руководителей разных уровней.

8. Уменьшение общего числа сбоев и нестандартных ситуаций, путем накопления информации по возникающим проблемам, дальнейшей ее автоматизированной обработке и получению рекомендаций по внесению изменений в состав ИТ с целью повышения отказоустойчивости.

9. Уменьшение времени восстановления системы после возникновения аварийной ситуации, путем оптимального построения подсистемы, ответственной за такие процессы восстановления, автоматизацией процесса диагностики и поиска источника проблем, введения возможности контроля за процессом восстановления и создания четко регламентированных процедур восстановления.

10. Качественное изменение уровня системы поддержки пользователей. Наличие единого оператора службы поддержки пользователей ИТ-ресурсами позволит не только зафиксировать проблему, но и проконтролировать время ее устранения персоналом ИТ-подразделения, а в дальнейшем - проанализировать причины возникновения проблемы и предложить способы устранения возможности ее возникновения. При этом во многих случая оператор сам сможет принять меры по разрешению проблемы.

11. Увеличение возможности самообразования пользователей. Сбор и структуризация наиболее традиционных, часто встречающихся вопросов по эксплуатации различных приложений, в единое хранилище для облегчения поиска и с подробными объяснениями способов решения вопроса будут размещаться на общедоступном ресурсе.

12. Осуществление полной инвентаризации имеющегося оборудования и технологий, учет эффективности его использования и контроль текущего состояния. Зная реальную загруженность дорогостоящего оборудования при необходимости можно переходить на более экономичные решения в случае, когда реальный процент использования дорогого оборудования не высок.

13. Прогнозирование возможных проблем и своевременное вырабатывание предложений по их предотвращению. Анализ тенденций, оценка потребностей в устройствах хранения, вычислительных мощностях и т.д. позволяет предвидеть возможную нехватку ресурсов и предусмотреть варианты решения проблем.

14. Оценка реальной стоимости отдельных ИТ-сервисов и всей ИТ-инфраструктуры в целом. Достигается путем четкой формализации предоставляемых услуг, полной прозрачности ИТ-инфраструктуры и возможности учета стоимости всех ее элементов, включая деятельность персонала.

15. Систематизация использования ПО. Для этого необходимо построить стандарты комплектования рабочих станций ПО и тем самым снизить издержки на их поддержку и сопровождение

**13. Библиотека ITIL. Предоставление услуг в ITIL. Поддержка услуг в ITIL.**

**ITIL –** библиотека, содержащая набор руководств по управлению, отладке и постоянного улучшения бизнес-процессов, связанных с ИТ.

С течением времени акцент переместился с разработки ИТ-приложений на управление ИТ-услугами. ИТ-приложения лишь тогда способствуют достижению корпоративных целей, когда система доступна пользователям, и при возникновении ошибок и необходимости модификации поддержка может быть оказана службой сопровождения. Первая редакция в 80-х годах (1986-1989), опубликована 1992, V30 2007, V4 последняя.

ITIL создавалась для систематического и последовательного распространения передового опыта по управлению ИТ-услугами. Этот подход основывается на качестве услуг и разработке эффективных и рациональных процессов.

Определения ITIL:

• ITIL – строго регламентированная система требований и рекомендаций по организации деятельности по управлению предоставлением информационных сервисов (сервисов ИТ или услуг ИТ) в соответствии с определенным уровнем качества и затрат.

• ITIL – не продукт, не программа, не система. ITIL – это методология, позволяющая обеспечить эффективное функционирование служб ИТ, удовлетворение нужд бизнес-пользователей, стабильное и предсказуемое развитие информационной системы.

В ITIL рассматриваются типовые модели, которые описывают цели, основные активности, входы и выходы разнообразных процессов, подлежащих внедрению в ИТ-подразделениях.

Библиотека несет в себе определенную новизну. Прослеживание роли ИТ-инфраструктуры в формировании прибавочной стоимости, соотнесение затрат на поддержание и развитие ИТ-инфраструктуры с реальными, а не абстрактными потребностями бизнеса, определение ответственности не только за функционирование отдельных технических единиц, а за весь предоставляемый сервис - вот наиболее значимые моменты отношений между бизнесом и ИТ-подразделениями.

ITIL – это тщательное описание модели жизненного цикла ИТ-подразделения, примеры реализации и комментарии специалистов.

Подход ITIL заключается в разделении процесса управления ИТ на несколько дисциплин, каждая из которых нацелена на решение определенных задач (функций), но во взаимодействии с остальными, охватывают такие области, как:

• управление изменениями (Change Management);

• управление проблемами (Problem Management);

• управление конфигурацией (Configuration management);

• управление уровнем сервиса (Service Level Management);

• управление планированием (Capacity Management);

• управление стоимостью (Cost Management) и множество других.

Ключевая концепция – управление сервисом, на которой сфокусирована вся методология и объединяющая все остальные компоненты для достижения цели – выполнения сервисных соглашений с пользователями.

Управление сервисом включает в себя множество процедур, позволяющих быстро и эффективно формулировать, изменять и контролировать определенные для каждого пользователя уровни сервиса по заранее заданным критериям и параметрам функционирования системы.

ITIL предлагает структурированную основу для организации всех видов деятельности в ИТ-подразделении, являющихся частью оказания ИТ-услуг. Эти виды деятельности образуют процессы, составляющие основу для развития степени зрелости ИТ Сервис-менеджмента. Каждый из этих процессов решает одну или несколько задач ИТ-департамента, такие, как разработка услуг (сервисов), управление Инфраструктурой, **предоставление и поддержка услуг**.

Преимущества заказчиков/пользователей:

• предоставление ИТ-услуг становится более ориентированным на заказчика, соглашения о качестве услуг способствуют улучшению взаимоотношений;

• услуги описываются лучше, на языке заказчика и с требуемой детализацией;

• лучше контролируются качество и стоимость услуг;

• улучшается взаимосвязь компании с ИТ-организацией за счет определения точек контактов

Преимущества для ИТ-организаций:

• становится четко понятна структура ИТ-департамента, его организация становится более рациональной и более ориентированной на корпоративные цели;

• руководство организацией становится более целенаправленным, облегчается Управление Изменениями;

• эффективная структура процессов создает основу для эффективного аутсорсинга элементов ИТ-услуг;

• следование передовому опыту ITIL способствует изменению корпоративной культуры в направлении осознания, что задачей ИТ-департамента является предоставление услуг, и поддерживает внедрение системы обеспечения качества на основе стандартов серии ISO-9000;

• библиотека ITIL предоставляет единую «систему координат» и понятий для взаимодействия как в компании, так и с поставщиками, необходимую при разработке и стандартизации корпоративных процедур.

**14. Что понимается под открытыми системами. Свойства открытых систем. Структура среды ИС. Эталонная модель открытых систем. Протокол. Интерфейс. Инкапсуляция. Основные элементы модели взаимодействия открытых систем.**

**Открытая система** – система реализующая открытые спецификации на интерфейсы, службы и форматы данных достаточные для того чтобы обеспечить:

1. Возможность переноса (мобильность) прикладных систем, разработанных с минимальными изменениями на широкий диапазон систем.

2. Совместную работу (интероперабельность) с другими прикладными системами на локальных и удаленных платформах.

3. Взаимодействие с пользователями в стиле, облегчающем переход от системы к системе.

Общие **свойства открытых** ИС:

• расширяемость/масштабируемость (возможность добавления новых функций ИС или изменения некоторых уже имеющихся при неизменных остальных функциональных частях ИС)

• мобильность/переносимость (возможность переноса программ и данных при модернизации или замене аппаратных платформ ИС и возможности работы с ними специалистов, пользующихся ИТ, без их переподготовки при изменениях ИС)

• взаимодействие (способность к взаимодействию с другими ИС: технические средства, на которых реализована ИС, объединяются сетью или сетями различного уровня – от локальной до глобальной);

• стандартизуемость – ИС проектируются и разрабатываются на основе согласованных международных стандартов и предложений, реализация открытости осуществляется на базе функциональных стандартов (профилей) в области ИТ

• дружественность к пользователю – развитые унифицированные интерфейсы в процессах взаимодействия в системе "человек-машина" позволяют работать пользователю, не имеющему специальной "компьютерной" подготовки.

**Взаимодействующие части структура среды ИС**:

1. функциональная часть, включающая прикладные программы, которые реализуют функции прикладной области;

2. среда или системная часть, обеспечивающая исполнение прикладных программ.

С этим разделением тесно связаны две группы вопросов стандартизации:

1. стандарты интерфейсов взаимодействия прикладных программ со средой ИС, прикладной программный интерфейс API;

2. стандарты интерфейсов взаимодействия самой ИС с внешней для нее средой (EEI). Эти две группы интерфейсов определяют спецификации внешнего описания среды ИС-архитектуру, с точки зрения конечного пользователя, проектировщика ИС, прикладного программиста, разрабатывающего функциональные части ИС.

Спецификации внешних интерфейсов среды ИС, спецификации интерфейсов взаимодействия между компонентами самой среды, – точные описания всех необходимых функций, служб и форматов определенного интерфейса. Совокупность таких описаний составляет **эталонную модель открытых систем** (Reference Open System Model). Используется более 30 лет и определяется системной сетевой архитектурой (SNA), предложенной IBM в 1974 году. Она основана на разбиении вычислительной среды на 7 уровней, взаимодействие между которыми описывается соответствующими стандартами и обеспечивает связь уровней вне зависимости от построения уровня в каждой конкретной реализации.



Достоинство – детальное описание связей в среде с точки зрения технических устройств и коммуникационных взаимодействий. Она не принимает в расчет взаимосвязь с учетом мобильности прикладного ПО. Эталонная модель среды открытых систем (OSE/RM) определяет разделение любой ИС на приложения (прикладные программы и программные комплексы) и среду, в которой эти приложения функционируют. Между приложениями и средой определяются стандартизованные интерфейсы (API), которые являются необходимой частью профилей любой открытой системы. В профилях ИС могут быть определены унифицированные интерфейсы взаимодействия функциональных частей друг с другом и интерфейсы взаимодействия между компонентами среды ИС.

В стандарте ISO 7498 выделено семь уровней (слоев) информационного взаимодействия, которые отделены друг от друга стандартными интерфейсами:

1. уровень приложения (прикладной уровень)

2. уровень представления

3. сеансовый (уровень сессии)

4. транспортный

5. сетевой

6. канальный

7. физический.

В соответствии с этим, информационное взаимодействие двух или более систем представляет собой совокупность информационных взаимодействий уровневых подсистем, причем каждый слой локальной ИС взаимодействует, как правило, с соответствующим слоем удаленной системы. Взаимодействие осуществляется при помощи соответствующих протоколов связи и интерфейсов. Применяя методы инкапсуляции, можно использовать одни и те же программные модули на различных уровнях.

**Протокол –** набор алгоритмов (правил) взаимодействия объектов одноименных уровней различных систем.

**Интерфейс** **–** совокупность правил, в соответствии с которыми осуществляется взаимодействие с объектом данного или другого уровня. Стандартный интерфейс в некоторых спецификациях может называться услугой.

**Инкапсуляция** **–**процесс помещения фрагментированных блоков данных одного уровня в блоки данных другого уровня. Каждый уровень имеет протокольную спецификацию (набор правил, управляющих взаимодействием равноправных процессов одного и того же уровня) и перечень услуг, которые описывают стандартный интерфейс с расположенным выше уровнем.

Каждый уровень использует услуги расположенного ниже уровня, каждый расположенный ниже предоставляет услуги расположенному выше.

**Основные элементы** модели взаимодействия открытых систем:

1. Уровень

2. Объекты

3. Соединения

4. Физические средства соединения

**15. Средства анализа и управления сетями. Стандарт Telecommunication Management Network.**

**Средства анализа и управления сетями**

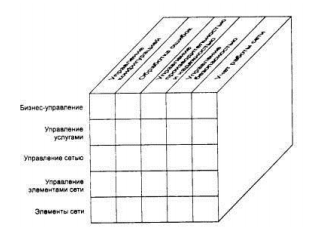
Распределенный характер крупной корпоративной сети делает невозможным поддержание ее работы без централизованной системы управления, которая в автоматическом режиме собирает информацию о состоянии каждого концентратора, коммутатора, мультиплексора и маршрутизатора и предоставляет эту информацию оператору сети. Обычно система управления работает в автоматизированном режиме, выполняя наиболее простые действия по управлению сетью автоматически, а сложные решения предоставляя принимать человеку на основе подготовленной системой информации. Система управления должна быть интегрированной. Это означает, что функции управления разнородными устройствами должны служить общей цели обслуживания конечных пользователей сети с заданным качеством.

Многоуровневое представление задач управления

Корпоративная сеть строится иерархически, отражая иерархию самого предприятия и его задач. Нижний уровень сети составляют элементы сети - отдельные компьютеры, коммуникационные устройства, каналы передачи данных. На следующем уровне иерархии эти элементы образуют сети разного масштаба - сеть рабочей группы, сеть отдела, сеть отделения и, наконец, сеть предприятия в целом. Для построения интегрированной системы управления разнородными элементами сети естественно применить многоуровневый иерархический подход.

**Telecommunication Management Network (TMN)**, разработанный совместными усилиями ITU-T, ISO, ANSI и ETSI. Хотя этот ориентация на использование общих принципов делает его полезным для построения любой крупной интегрированной системы управления сетями. Стандарты TMN состоят из большого количества рекомендаций ITU-T (и стандартов других организаций), но основные принципы модели TMN описаны в рекомендации М.3010.

На каждом уровне иерархии модели TMN решаются задачи одних и тех же пяти функциональных групп, (управления конфигурацией, производительностью, ошибками, безопасностью и учетом), однако на каждом уровне эти задачи имеют свою специфику. Модель TMN упрощенно можно представить в виде двухмерной диаграммы.



Нижний уровень - уровень элементов сети (Network Element layer, NE) - состоит из отдельных устройств сети: каналов, усилителей, оконечной аппаратуры, мультиплексоров, коммутаторов и т. п. Элементы могут содержать встроенные средства для поддержки управления - датчики, интерфейсы управления, а могут и представлять вещь в себе, требующую для связи с системой управления разработки специального оборудования - устройств связи с объектом, УСО. Современные технологии обычно имеют встроенные функции управления, которые позволяют выполнять хотя бы минимальные операции по контролю за состоянием устройства и за передаваемым устройством трафиком. Подобные функции встроены в технологии FDDI, ISDN, frame relay, SDH. В этом случае устройство всегда можно охватить системой управления, даже если оно не имеет специального блока управления, так как протокол технологии обязывает устройство поддерживать некоторые функции управления. Устройства, которые работают по протоколам, не имеющим встроенных функций контроля и управления, снабжаются отдельным блоком управления, который поддерживает один из двух наиболее распространенных протоколов управления - SNMP или CMIP. Эти протоколы относятся к прикладному уровню модели OSI.

Следующий уровень - уровень управления элементами сети (network element management layer) - представляет собой элементарные системы управления. Элементарные системы управления автономно управляют отдельными элементами сети - контролируют канал связи SDH, управляют коммутатором или мультиплексором. Уровень управления элементами изолирует верхние слои системы управления от деталей и особенностей управления конкретным оборудованием. Этот уровень ответственен за моделирование поведения оборудования и функциональных ресурсов нижележащей сети. Атрибуты этих моделей позволяют управлять различными аспектами поведения управляемых ресурсов. Обычно элементарные системы управления разрабатываются и поставляются производителями оборудования. Примерами таких систем могут служить системы управления CiscoView от Cisco Systems, Optivity от Bay Networks, RADView от RAD Data Communications и т. д.

Выше лежит уровень управления сетью (Network management layer). Этот уровень координирует работу элементарных систем управления, позволяя контролировать конфигурацию составных каналов, согласовывать работу транспортных подсетей разных технологий и т. п. С помощью этого уровня сеть начинает работать как единое целое, передавая данные между своими абонентами.

Следующий уровень - уровень управления услугами (Service management layer) - занимается контролем и управлением за транспортными и информационными услугами, которые предоставляются конечным пользователям сети. В задачу этого уровня входит подготовка сети к предоставлению определенной услуги, ее активизация, обработка вызовов клиентов. Формирование услуги (service provisioning) заключается в фиксации в базе данных значений параметров услуги, например, требуемой средней пропускной способности, максимальных величин задержек пакетов, коэффициента готовности и т. п. В функции этого уровня входит также выдача уровню управления сетью задания на конфигурирование виртуального или физического канала связи для поддержания услуги. После формирования услуги данный уровень занимается контролем за качеством ее реализации, то есть за соблюдением сетью всех принятых на себя обязательств в отношении производительности и надежности транспортных услуг. Результаты контроля качества обслуживания нужны, в частности, для подсчета оплаты за пользование услугами клиентами сети. Например, в сети frame relay уровень управления услугами следит за заказанными пользователем значениями средней скорости CIR и согласованной пульсации Вс, фиксируя нарушения со стороны пользователя и сети.

**16. Архитектуры систем управления сетями. Схема менеджер – агент. Структуры распределенных систем управления. Одноранговая и иерархическая модели управления сетями.**

**Архитектуры систем управления сетями**

Выделение в системах управления типовых групп функций и разбиение этих функций на уровни еще не дает ответа на вопрос, каким же образом устроены системы управления, из каких элементов они состоят и какие архитектуры связей этих элементов используются на практике.

**Схема менеджер – агент**

• Агент – устройство или программа, которые устанавливаются в элементах компьютерной сети для централизованного управления этими элементами и всей сетью. Агенты являются частью системы сетевого управления. Каждый агент собирает данные и управляет определенным элементом сети. Менеджеры (серверы) системы управления, собирают данные от своих агентов, обобщают их и хранят в бд.

• Менеджер – основная управляющая программа, которая устанавливается на отдельном компьютере, который выполняет роль консоли управления для оператора или администратора системы. Менеджер взаимодействует с агентами по стандартному протоколу, например, SNMP. Агент является посредником между управляемым ресурсом и основной управляющей программой-менеджером.

Агент поставляет менеджеру обработанную и представленную в нормализованном виде информацию, на основе которой менеджер принимает решения по управлению, а также выполняет дальнейшее обобщение данных о состоянии управляемого ресурса, например, строит зависимость загрузки порта от времени. Операторы, работающие за рабочими станциями, могут соединиться с любым из менеджеров и с помощью графического интерфейса просмотреть данные об управляемой сети, а также выдать менеджеру некоторые директивы по управлению сетью или ее элементами. В основе любой системы управления сетью лежит элементарная схема взаимодействия агента с менеджером. На основе этой схемы могут быть построены системы практически любой сложности с большим количеством агентов и менеджеров разного типа. Схема:



Модель менеджер-агент лежит в основе таких популярных стандартов управления, как стандарты Internet на основе протокола SNMP и стандарты управления ISO/OSI на основе протокола CMIP.

Агенты могут отличаться различным уровнем интеллекта – они могут обладать как самым минимальным интеллектом, необходимым для подсчета проходящих через оборудование кадров и пакетов, так и весьма высоким, достаточным для выполнения самостоятельных действий по выполнению последовательности управляющих действий в аварийных ситуациях, построению временных зависимостей, фильтрации аварийных сообщений и т. п.

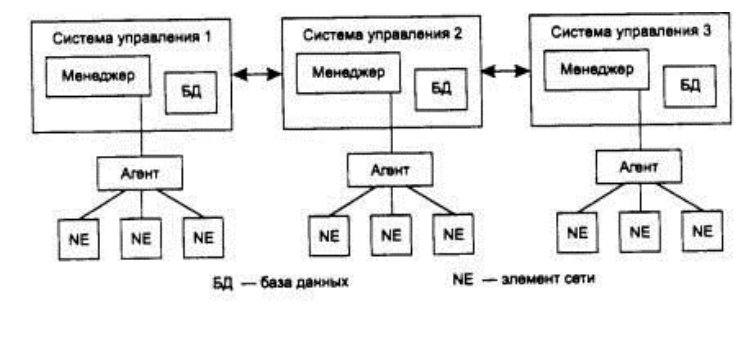
**Структуры распределенных систем управления**

В крупной корпоративной сети полностью централизованная система управления, построенная на базе единственного менеджера, может плохо работать по причинам:

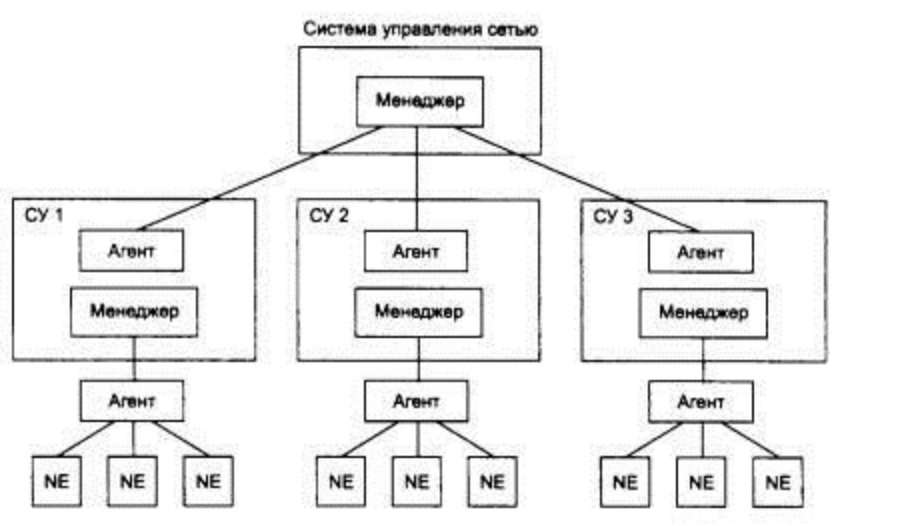
* такой вариант не обеспечивает необходимой масштабируемости по производительности, так как единственный менеджер вынужден будет обрабатывать весь поток сообщений от всех агентов, что при нескольких тысячах управляемых объектов потребует очень высокопроизводительной платформы для работы менеджера и перегрузит служебной управляющей информацией каналы передачи данных в той сети, где будет расположен менеджер.
* не обеспечит необходимого уровня надежности, т.к. при отказе единственного менеджера будет потеряно управление сетью.
* в большой распределенной сети целесообразно располагать в каждом географическом пункте отдельным оператором или администратором, управляющим своей частью сети, а это удобнее реализовать с помощью отдельных менеджеров для каждого оператора.

Схема «менеджер-агент» позволяет строить достаточно сложные в структурном отношении распределенные системы управления. Обычно распределенная система управления включает большое количество связок менеджер-агент, которые дополняются рабочими станциями операторов сети, с помощью которых они получают доступ к менеджерам. Наличие нескольких менеджеров позволяет распределить между ними нагрузку по обработке данных управления, обеспечивая масштабируемость системы. Чаще всего используются два подхода – **одноранговый и иерархический**.

В одноранговых связях каждый менеджер управляет своей частью сети на основе информации, получаемой от нижележащих агентов. Центральный менеджер отсутствует. Координация работы менеджеров достигается за счет обмена информацией между бд каждого менеджера. Одноранговое построение системы управления сегодня считается неэффективным и устаревшим:



Гораздо более гибким является иерархическое построение связей между менеджерами. Каждый менеджер нижнего уровня выполняет функции агента для менеджера верхнего уровня. Такой агент работает уже с гораздо более укрупненной моделью (MIB) своей части сети, в которой собирается именно та информация, которая нужна менеджеру верхнего уровня для управления сетью в целом. Обычно для разработки моделей сети на разных уровнях проектирование начинают с верхнего уровня, на котором определяется состав информации, требуемой от менеджеров-агентов более низкого уровня, поэтому такой подход назван подходом «сверху вниз». Он сокращает объемы информации, циркулирующей между уровнями системы управления, и приводит к гораздо более эффективной системе управления:



**17. Управление конфигурацией. Обзор метода. Основные термины. Функции управления конфигурациями. Процедура управления конфигурациями. Основные процессы.**

**Управление** **конфигурациями** определяется как процесс, с помощью которого администрация программы или проекта может систематически идентифицировать, устанавливать связи, сопровождать и управлять различными компонентами программы или проекта.

Этот процесс гарантирует целостность компонент и трассируемость всех изменений, возникающих в любой момент жизненного цикла программы или проекта.

Управление конфигурациями позволяет команде разработчиков программы или проекта точно определять статус любой компоненты во все время ее жизненного цикла и позволяет перевоссоздать любую версию в любой момент времени. Компонентами могут быть любые комбинации аппаратуры, программ, обслуживания и обучения.

**Назначение метода**

Метод управления конфигурациями обеспечивает порядок, в котором применяются техническое и административное руководство, и наблюдение.

Назначение методов:

1. Идентифицировать, определить и документировать объекты и схемы конфигураций

2. Прослеживать и управлять изменениями объектов и схем конфигураций

3. Запоминание и обработка изменений отчетов и статуса реализаций.

Так как применение этого метода требует от команды разработчиков программы или проекта большого объема взаимосвязей, члены команды должны проявить максимум общечеловеческих достоинств, таких как такт и способность договариваться. В связи с этим управление конфигурациями можно рассматривать как искусство и, точно так же, как техническую дисциплину.

Этот метод предлагает достаточное количество деталей, которые полезны для больших, сложных программ или проектов. Например, в качестве примера предлагается большое число трассировочных записей. При более ограниченных потребностях метод может быть адаптирован к более ограниченному варианту

**Основные термины:**

Объектами конфигурации являются все основные результаты деятельности по разработке программы или проекта. Такие результаты идентифицируются и управляются с помощью управления конфигурациями.

Объекты конфигурации могут включать

– аппаратуру,

– программы,

– документацию,

– обучение и обслуживание.

Объектам конфигурации могут быть присвоены номера для простоты прослеживаний. Заметим, что термин "компонент" используется в этом документе в очень обобщенном смысле, связываясь с некоторой частью общего решения. Термин "система" используется для обозначения полного решения, которое может включать программы, аппаратуру и другие компоненты.

Процесс управления изменениями определяет оценку и трассировку запросов на изменения, анализ потенциального влияния изменений и принятие решений по внесению изменений в объекты конфигурации. Между управлением изменениями и управлением конфигурациями должен быть эффективный интерфейс. Реализация запросов на изменения должна включать трассировку данного изменения через процесс управления конфигурациями.

Трассируемостью называется возможность идентифицировать и историю, и текущее состояние (статус) каждого объекта конфигурации в любой точке жизненного цикла программы или проекта. Необходимой также является и возможность трассировки объектов конфигурации относительно требований заказчика, как они изложены в Спецификации Требований.

**Функции управления конфигурациями**

Организация управления конфигурациями может быть поделена на несколько функций со специфическими обязанностями.

Этими функциями являются:

• Административный контроль и планирование конфигураций: Ответственные за эту функцию специалисты обязаны обеспечить написание процедур, сопровождение разнообразных файлов управления конфигурациями и модификацию (по мере необходимости) записей ВСК.

• Управление данными: Эта функция управляет и прослеживает всю программную/проектную корреспонденцию и внешнюю документацию. Управление данными также контролирует все номера, присваиваемые документам, программам и аппаратуре.

• Библиотеки: В принципе могут быть выделены программные библиотекари, библиотекари документации и аппаратные библиотекари, в зависимости от типов объектов, определения которых должны быть запомнены в управляемой памяти. Нести ответственность за реализацию этой функции может как один человек, так и несколько.

• Управление программными конфигурациями: Эта функция реализует управление всеми разработанными кодами и всей документацией, связанной с разработкой программ.

• Управление аппаратными конфигурациями: Эта функция реализует управление всей аппаратурой. Последняя может включать покупное оборудование.

Процесс управления конфигурациями состоит из следующих подпроцессов:

1. Идентификация конфигураций

2. Контроль за конфигурациями

3. Вычисление статуса конфигурации

4. Аудиты/обзоры конфигураций

Фундаментальным критерием применения процедур является следующее условие: Может ли "судьба" каждого объекта конфигурации быть прослежена в течении всего жизненного цикла программы или проекта?

1. Идентификация конфигураций

Процесс идентификации конфигураций требует выполнения следующих шагов:

1. Определение объектов конфигурации

2. Выбор схемы наименования объектов конфигурации

3. Определение утверждаемых схем

4. Определение внутренних схем

**18. Безопасность ИС. Основные аспекты информационной безопасности. Виды и группы угроз ИБ, классы отказов.**

**Безопасность ИС** – свойство, заключающееся в способности системы обеспечить конфиденциальность и целостность информации, т.е. защиту информации от НСД с целью ее раскрытия, изменения или разрушения.

Аспекты ИБ:

• доступность (возможность за приемлемое время получить требуемую информационную услугу);

• целостность (актуальность и непротиворечивость информации, ее защищенность от разрушения и несанкционированного изменения);

• конфиденциальность (защита от несанкционированного ознакомления).

Главная цель мер, предпринимаемых на административном уровне, состоит в том, чтобы сформировать программу работ в области повышения доступности информационных сервисов и обеспечить ее выполнение, выделяя необходимые ресурсы и контролируя фактическое состояние дел. Первым этапом выработки подобной программы является анализ угроз и рисков.

Группы угроз:

1. Угроза раскрытия – возможность того, что информация станет известной тому, кому не следовало бы ее знать.

2. Угроза целостности – умышленное несанкционированное изменение (модификация или удаление) данных, хранящихся в вычислительной системе или передаваемых из одной системы в другую.

3. Угроза отказа в обслуживании – возможность появления блокировки доступа к некоторому ресурсу вычислительной системы.

Выделяют следующие **классы отказов**:

* 1. Отказ пользователей – возникает по следующим причинам:

• нежелание работать с ИС;

• невозможность работать с системой в силу отсутствия соответствующей подготовки;

• невозможность работать с системой в силу отсутствия технической поддержки.

2. Внутренний отказ ИС, причины:

• отступление (случайное или умышленное) от установленных правил эксплуатации;

• ошибки при (пере)конфигурировании системы;

• отказы программного и аппаратного обеспечения;

• разрушение данных;

• разрушение или повреждение аппаратуры.

1. Отказ поддерживающей инфраструктуры, причины:

• нарушение работы (случайное или умышленное) систем связи, электропитания, водоснабжения, кондиционирования;

• разрушение или повреждение помещений;

• невозможность или нежелание выполнения обслуживающим персоналом и/или пользователями своих обязанностей (гражданские беспорядки, аварии на транспорте, террористический акт или его угроза, забастовка и т.п.).

По природе возникновения угрозы можно разделить на:

• Естественные угрозы – угрозы, связанные с воздействиями на ИС объективных физических процессов или природных явлений.

• Искусственные угрозы – угрозы ИС, связанные с деятельностью человека.

**19. Непреднамеренные и преднамеренные действия пользователей, предоставляющие угрозу безопасности ИС. Модель нарушителя ИБ. Типы нарушителей, классификация. Мотивы нарушений.**

Пользователем ИС могут быть осуществлены следующие **непреднамеренные** действия, представляющие угрозу безопасности ИС:

– доведение до состояния частичного или полного отказа системы, разрушение аппаратных, программных, информационных ресурсов системы (порча оборудования, носителей информации, удаление, искажение файлов с важной информацией или программ);

– неправомерное включение оборудования или изменение режимов работы устройств и программ;

– запуск сервисных программ, способных при некомпетентном использовании вызывать потерю работоспособности системы или необратимые изменения в системе;

– нелегальное внедрение и использование неучтенных программ, не являющихся необходимыми для выполнения служебных обязанностей, с последующим необоснованным расходованием ресурсов (загрузка процессора, захват оперативной памяти и памяти внешних носителей);

– заражение компьютера вирусами;

– разглашение конфиденциальной информации;

– разглашение, передача или утрата атрибутов разграничения доступа (паролей, ключей шифрования, идентификационных карточек, пропусков и т. п.);

– игнорирование организационных ограничений;

– некомпетентное использование, настройка или неправомерное отключение средств защиты информации;

– пересылка данных по ошибочному адресу абонента;

– ввод ошибочных данных;

– повреждение каналов связи.

**Преднамеренные**:

– физическое разрушение системы или вывод из строя наиболее важных ее компонентов;

– отключение или вывод из строя подсистем обеспечения функционирования вычислительных систем (электропитания, охлаждения и вентиляции, линий связи и т. п.);

– дезорганизация функционирования системы (изменение режимов работы устройств или программ, создание мощных активных радиопомех и т. п.);

– внедрение агентов в число персонала, вербовка персонала или отдельных пользователей, имеющих определенные полномочия;

– применение подслушивающих устройств, дистанционная фото и видеосъемка и т. п.;

– перехват побочных электромагнитных, акустических и других излучений устройств и линий связи;

– перехват данных, передаваемых по каналам связи, и их анализ с целью осуществления попыток проникновения в систему;

– хищение носителей информации;

– несанкционированное копирование носителей информации;

– хищение производственных отходов (распечаток, записей, списанных носителей информации и т.п.);

– чтение остатков информации из оперативной памяти и с внешних запоминающих устройств, чтение информации из областей оперативной памяти, используемых ОС;

– незаконное получение паролей и других реквизитов разграничения доступа (агентурным путем, используя халатность пользователей, путем подбора, имитации интерфейса системы и т. п.) с последующей маскировкой под зарегистрированного пользователя;

– несанкционированное использование терминалов пользователей, имеющих уникальные физические характеристики;

– вскрытие шифров криптозащиты информации;

– внедрение аппаратных спецвложений, программ «закладок» и «троянских коней».

**Модель нарушителя** – формализованное описание или представления комплекса возможностей нарушителя по реализации тех или иных угроз безопасности информации, при разработке которой делаются предположения:

– о категориях лиц, к которым может принадлежать нарушитель;

– о мотивах действий нарушителя;

– о квалификации нарушителя и его технической оснащенности;

– о характере возможных действий нарушителя.

**Типы нарушителей по отношению к ИС:**

Внутренние нарушители. Лица из категорий персонала:

• пользователи системы;

• персонал, обслуживающий технические средства (инженеры, техники);

• сотрудники отделов разработки и сопровождения ПО (прикладные и системные программисты);

• технический персонал, обслуживающий здание (уборщики, электрики, сантехники и другие сотрудники, имеющие доступ в здание и помещения, где расположены компоненты ИС);

• сотрудники службы безопасности ИС;

• руководители различных уровней должностной иерархии.

Внешние нарушители. Посторонние лица:

• клиенты;

• посетители;

• представители организаций, взаимодействующих по вопросам обеспечения жизнедеятельности организации (энерго-, водо-, теплоснабжение и т. п.);

• представители конкурирующих организаций или лица, действующие по их заданию;

• лица, случайно или умышленно нарушившие пропускной режим (без цели нарушения безопасности ИС).

Нарушители **классифицируются** по следующим признакам:

1. По уровню знаний об ИС.

2. По уровню возможностей - нарушители определяются, как:

• применяющие чисто агентурные методы получения сведений;

• применяющие пассивные средства (технические средства перехвата без модификации компонентов системы);

• использующие только штатные средства и недостатки систем защиты для ее преодоления, а также компактные магнитные носители информации, которые могут быть скрытно пронесены через посты охраны;

• применяющие методы и средства активного воздействия (модификация и подключение дополнительных механических средств, подключение к каналам передачи данных, внедрение программных «закладок» и использование специальных инструментальных и технологических программ).

3. По месту действия - нарушители могут быть:

• не имеющие доступа на контролируемую территорию организации;

• действующие с контролируемой территории без доступа в здания и сооружения;

• действующие внутри помещений, но без доступа к техническим средствам ИС;

• действующие с рабочих мест конечных пользователей ИС;

• имеющие доступ в зону данных (баз данных, архивов и т. п.);

• имеющие доступ в зону управления средствами обеспечения безопасности ИС.

Можно выделить три основных **мотива нарушений**:

• безответственность;

• самоутверждение;

• корыстный интерес.

**20.Системы защиты ИС. Основные методы защиты информации. Средства обеспечения ИБ. Модели администрирования сети и способы обеспечения безопасности (централизованное, распределенное, администрирование по ОС, смешанная модель).**

**Система защиты** – это совокупность специальных мер правового и административного характера, организационных мероприятий, программно-аппаратных средств защиты, а также специального персонала, предназначенных для обеспечения информационной безопасности.

Для построения эффективной системы защиты необходимо провести следующие работы:

• определить угрозы безопасности информации;

• выявить возможные каналы утечки информации и несанкционированного доступа (НСД) к данным;

• построить модель потенциального нарушителя;

• выбрать соответствующие меры, методы, механизмы и средства зашиты.

Основные **методы защиты** информации:

1. Создание препятствий – методы физического преграждения злоумышленнику пути к защищаемой информации (аппаратуре, носителям информации и т. д.).

2. Управление доступом – метод защиты информации регулированием использования всех ресурсов компьютерной информационной системы (элементов баз данных, программных и технических средств).

3. Защита от НСД к ресурсам компьютера – это комплексная проблема, подразумевающая решение следующих вопросов:

– присвоение пользователю, терминалам, программам, файлам и каналам связи уникальных имен и кодов (идентификаторов);

– выполнение процедур установления подлинности при обращениях к ИС, т.е. проверка того, что лицо или устройство, сообщившее идентификатор, в действительности ему соответствует;

– проверка полномочий, то есть проверка права пользователя на доступ к системе или запрашиваемым данным;

– автоматическая регистрация в специальном журнале всех как удовлетворенных, так и отвергнутых запросов к информационным ресурсам с указанием идентификатора пользователя, терминала, времени и сущности запроса, то есть ведение аудита.

4. Маскировка – метод защиты информации путем ее криптографического закрытия.

5. Регламентация – метод защиты информации, создающий такие условия автоматизированной обработки, хранения и передачи защищаемой информации, при которых возможности несанкционированного доступа к ней сводились бы к минимуму.

6. Принуждение – метод защиты, при котором пользователи и персонал системы вынуждены соблюдать правила обработки, передачи и использования защищаемой информации под угрозой материальной, административной или уголовной ответственности.

Рассмотренные методы обеспечения безопасности реализуются на практике за счет применения различных средств защиты.

**Средства обеспечения** ИБ по способу их реализации можно разделить на классы методов:

- аппаратные методы, реализующие физическую защиту системы от несанкционированного доступа, аппаратные функции идентификации периферийных терминалов системы и пользователей, режимы подключения сетевых компонентов и т. д.

- технические средствам физической защиты: механические, электронно-механические, электромеханические, оптические, акустические, лазерные, радио и другие устройства, системы и сооружения, предназначенные для создания физических препятствий на пути к защищаемой информации и способные выполнять самостоятельно или в комплексе с другими средствами функции защиты информации.

- организационные методы подразумевают рациональное конфигурирование, организацию и администрирование системы. В первую очередь это касается сетевых ИС, их ОС, полномочий сетевого администратора, набора обязательных инструкций, определяющих порядок доступа и работы в сети пользователей;

- технологические методы, включающие в себя технологии выполнения сетевого администрирования, мониторинга и аудита безопасности информационных ресурсов, ведения электронных журналов регистрации пользователей, фильтрации и антивирусной обработки поступающей информации;

- программные методы – самые распространенные методы (например, программы идентификации пользователей, парольной защиты и проверки полномочий, брандмауэры, криптопротоколы и т. д.). Без использования программной составляющей практически невыполнимы первые три группы методов (т.е. в чистом виде организационные, технологические и аппаратные методы защиты, реализованы быть не могут – все они содержат программный компонент).

**Модели администрирования сети и способы обеспечения безопасности**

Администрирование сети можно организовать одним из четырех основных способов:

• централизованно на всем предприятии;

• по отделам или группам («распределенное» администрирование);

• по ОС;

• в виде сочетания предыдущих способов.

Модели администрирования небольших и крупных, сложных систем могут совпадать. Они будут отличаться масштабами, но не по сути.

Централизованное администрирование. Один человек, группа или отдел занимается администрированием всей сети организации, ее пользователей и ресурсов. Минус: ее недостаточная масштабируемость, отсутствие отказоустойчивости. Производительность центрального компьютера всегда будет ограничителем количества пользователей, работающих с данным приложением, а отказ центрального компьютера приводит к прекращению работы всех пользователей. Эта модель хорошо подходит небольшим и средним организациям, но может оказаться медленной и неэффективной для крупного или географически разбросанного предприятия. Однако с точки зрения безопасности является наилучшим. Оно гарантирует, что системная политика и процедуры являются однообразными для всей организации.

Распределенное администрирование. Управление сетью осуществляется на уровне отдела или рабочей группы. Администрирование на этом уровне может быстро откликаться на нужды пользователей, часто это достигается за счет безопасности сети. При наличии нескольких администраторов политика администрирования в разных рабочих группах будет отличаться. Чем больше групп имеется в системе, тем больше доверительных отношений им требуется, что повышает возможность того, что в систему проникнет злоумышленник и воспользуется этими доверительными отношениями, чтобы добраться до совершенно секретной информации.

Администрирование по ОС. Средства обеспечения безопасности значительно различаются в зависимости от используемых ОС. Например, если имеется свой администратор у сервера Windows NТ Server, свой – у сервера Novell Net Wаге и свой – у систем UNIX, то администратор каждой системы будет сам обеспечивать ее безопасность. Однако потребуется кто-то, кто будет разрешать различия во мнениях администраторов в случае возникновения проблем.

Смешанная модель администрирования. Сочетает элементы централизованной и распределенной моделей. Центральный администратор (или группа) гарантирует проведение политики безопасности на всем предприятии, а администраторы на уровне отделов или рабочих групп выполняют повседневную работу. Требуется больше затрат на штат, чем может себе позволить небольшая организация, поэтому применение смешанной модели администрирования, ограничивается крупными предприятиями.

**21. Инсталляция информационной системы. Основные этапы. Домен. Имена доменов. Отношения доменов. Модели доменов**

Цель построения всякой системы — достижение состояния, при котором все имеющиеся объекты управления будут находиться под контролем и готовы адекватно реагировать на управляющие воздействия.

Домен — это группа компьютеров, которые совместно используют общую базу данных и общую политику безопасности. Он является основным организационным блоком сети Windows NT Server.

В сети, состоящей из двух и более доменов, каждый домен действует как отдельная сеть со своей базой данных учетных записей.

Однако даже в наиболее жестко структурированной организации некоторым пользователям из одного домена могут понадобиться какие-нибудь ресурсы из другого домена.

Обычное решение этой проблемы, связанной с настройкой уровней доступа пользователей между различными доменами, называется установлением доверительных отношений.

**Модель с одним доменом***-* самая простая модель; все серверы и клиенты входят в один домен. Локальные и глобальные группы совпадают, а все администраторы могут администрировать все серверы. Поскольку домен один, нет необходимости в доверительных отношениях.

**Модель с одним главным доменом** обеспечивает централизованное администри­рование пользователей наряду с логической группировкой ресурсов по подраз­делениям. В эту модель входят один или несколько доменов подразделений (доменов ресурсов — secondary domain), которые доверяют одному главному домену (master domain).

***Модель с несколькими главными доменам*** рекомендуется для организаций с большим числом пользователей и централизованной структурой управления. В ней обеспечивается централизованное администриро­вание двух и более главных доменов, а ресурсы распределены между доменами ресурсов.

**Модель с несколькими главными доменами и полностью доверительными отно­шениями**целесообразно использовать в организации со строгой, централизованной структурой управления, когда предпочтение отдается распределенному администрированию пользователей и ресурсов. В этом варианте между каждой парой доменов устанавливаются двусторонние доверительные отноше­ния, а учетная запись пользователя создается в том домене, в который он будет входить по умолчанию (домен, в котором хранится учетная запись пользова­теля, известен еще как основной домен — home domain). Домены ресурсов отвечают за управление собственными пользователями и глобальными группами, при этом к соответствующим учетным записям имеется доступ из сети

**2 блок**

**1. Определение БД. Определение СУБД. Расписать два уровня администрирования в СУБД. Функции администратора СУБД.**

**Ба́за да́нных**  — представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины

**Система управления базами данных (СУБД)** – это комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными. Иными словами, СУБД — это набор программ, позволяющий организовывать, контролировать и администрировать базы данных. Большинство сайтов не могут функционировать без базы данных, поэтому СУБД используется практически повсеместно.

**1 уровень - системный администратор (администратор СУБД)**

Его функции относятся ко всей системе, его права и привилегии распространяются на все БД и объекты и субъекты в них.

**2 уровень - администраторы базы данных (БД).**

Его функции относятся не только к подмножеству системных ресурсов, выделенных для конкретной БД, его права и привилегии распространяются на объекты конкретной БД и субъекты, имеющие доступ к ним.

**Функции администратора**

1. Инсталляция СУБД
2. Управление памятью
3. Управление разделением данных между пользователями
4. Копирование и восстановление БД
5. Управление безопасностью в системе
6. Передача данных между СУБД и другими системами
7. Управление производительностью

**2. Какие группы пользователей СУБД выделяют? Дискреционное управление доступом. Средства дискреционной защиты в СУБД. Мандатное управление доступом. Средства мандатной защиты в СУБД.**

1. Прикладные программисты – создают программы, использующие БД
2. Конечные пользователи – работают с непосредственно с БД.
3. Администраторы БД – непосредственное управленние, создание новых БД, технический контроль функционирования.

**Дискреционное управление доступам** (discretionary access control) — разграничение доступа между поименованными субъектами и поименованными объектами. Субъект с определенным правом доступа может передать это право любому другому субъекту. Дискреционная защита является многоуровневой логической защитой. Логическая защита в СУБД представляет собой набор привилегий или ролей по отношению к защищаемому объекту. К логической защите можно отнести и владение таблицей (представлением). Владелец таблицы может изменять (расширять, отнимать, ограничивать доступ) набор привилегий (логическую защиту). Данные о логической защите находятся в системных таблицах базы данных и отделены от защищаемых объектов (от таблиц или представлений).

**Мандатное управление доступом** (mandatory access control) — это разграничение доступа субъектов к объектам данных, основанное на характеризуемой меткой конфиденциальности информации, которая содержится в объектах, и на официальном разрешении (допуске) субъектов обращаться к информации такого уровня конфиденциальности. Для чего же нужна мандатная защита? Средства произвольного управления доступом характерны для уровня безопасности C. Как правило, их, в принципе, вполне достаточно для подавляющего большинства коммерческих приложений. Тем не менее они не решают одной весьма важной задачи — задачи слежения за передачей информации. Средства произвольного управления доступом не могут помешать авторизованному пользователю законным образом получить секретную информацию и затем сделать ее доступной для других, неавторизованных, пользователей. Нетрудно понять, почему это так. При произвольном управлении доступом привилегии существуют отдельно от данных (в случае реляционных СУБД — отдельно от строк реляционных таблиц), в результате чего данные оказываются «обезличенными» и ничто не мешает передать их кому угодно даже средствами самой СУБД; для этого нужно лишь получить доступ к таблице или представлению. Физическая защита СУБД главным образом характеризует данные (их принадлежность, важность, представительность и пр.). Это в основном метки безопасности, описывающие группу принадлежности и уровни конфиденциальности и ценности данных объекта (таблицы, столбца, строки или поля). Метки безопасности (физическая защита) неизменны на всем протяжении существования объекта защиты (они уничтожаются только вместе с ним) и территориально (на диске) располагаются вместе с защищаемыми данными, а не в системном каталоге, как это происходит при логической защите.

**3. Определение транзакции. Суть механизма транзакций? Основные уровни восстановления БД.**

**Транзакции** - атомарного действия над БД, переводящего ее из одного целостного состояния в другое целостное состояние. Другими словами, транзакция - это последовательность операций, которые должны быть или все выполнены или все не выполнены (все или ничего). (Плоские, цепочные, вложенные)

Транзакция - это механизм, позволяющий объединять различные действия в логические блоки и обеспечить возможность принимать решения об успешности действий всего блока операций в целом

1. Механизм транзакций обязательно используется для ВСЕХ операций в базе данных (о некоторых особых случаях будет рассказано ниже). Возможно, некоторые разработчики, пользующиеся высокоуровневыми инструментами разработки приложений баз данных, могут заявить, что они никогда не применяли транзакции и не видят в них нужды. Но это всего лишь означает, что всю работу по управлению транзакциями брал на себя инструмент разработки (и вряд ли он управлял ими достаточно эффективно!).
2. Сочетание слов "логический блок" напоминает нам, что транзакции изначально задумывались и реализовывались как механизм управления бизнес- логикой в базах данных. Это означает, что объединением некоторой последовательности операций в транзакцию управляет клиентское приложение базы данных (а в конечном итоге - пользователь).
3. Подтверждение или отмена результатов операций, объединенных одной транзакцией, не означает, что все эти операции выполнились успешно (или закончились ошибкой). Подтверждение транзакции - это решение о том, что следует оставить в базе данных результаты работы всех операций, входящих в транзакцию, вне зависимости от того, как они закончились. Если клиентское приложение (фактически человек-пользователь) решило подтвердить результаты транзакции, то при этом подтвердятся результаты всех успешных действий (а у неуспешных действий просто не будет результатов, поэтому произойдет лишь формальное подтверждение). Если клиентское приложение решило откатить транзакцию, то все результаты всех действий и успешные, и неуспешные, будут аннулированы.

**Уровни восстановления**

* 1. Оперативное восстановление, которое характеризуется возможностью восстановления на уровне отдельных транзакций при ненормальном окончании ситуации манипулирования данными.
  2. Промежуточное восстановление – восстановление всех выполняемых на момент сбоя транзакций.
  3. Длительное восстановление – при разрушении БД в результате дефекта на диске восстановление осуществляется с помощью копии БД. Затем воспроизводит результаты выполненных после снятия копии транзакций, возвращает систему в состояние на момент разрушения.

**4. Языковые средства разграничения доступа. Основные команды языка SQL. Концепция и реализация механизма ролей БД.**

Составной частью языка SQL является язык, позволяющий описывать устанавливать те или иные назначения доступа и другие необходимые установки политики безопасности в конкретной АИС.

**SHOW DATABASES**

SQL-команда, которая отвечает за просмотр доступных баз данных.

**CREATE DATABASE**

Команда для создания новой базы данных.

**USE**

С помощью этой SQL-команды USE <database\_name> выбирается база данных, необходимая для дальнейшей работы с ней.

**SOURCE**

А SOURCE <file.sql> позволит выполнить сразу несколько SQL-команд, содержащихся в файле с расширением .sql.

**DROP DATABASE**

Стандартная SQL-команда для удаления целой базы данных.

SHOW TABLES

С помощью этой несложной команды можно увидеть все таблицы, которые доступны в базе данных.

**CREATE TABLE**

SQL-команда для создания новой таблицы

**DESCRIBE**

С помощью DESCRIBE <table\_name> можно просмотреть различные сведения (тип значений, является ключом или нет) о столбцах таблицы.

**INSERT**

Команда INSERT INTO <table\_name> в SQL отвечает за добавление данных в таблицу

**UPDATE**

SQL-команда для обновления данных таблицы

**DELETE**

SQL-команда DELETE FROM <table\_name> используется для удаления данных из таблицы.

**DROP TABLE**

А так можно удалить всю таблицу целиком.

**SELECT**

Далее мы рассмотрим основные команды SQL, которые позволяют работать непосредственно с данными. К одной из таких SQL-команд относится SELECT для получения данных из выбранной таблицы

**SELECT DISTINCT**

В столбцах таблицы могут содержаться повторяющиеся данные. Используйте SELECT DISTINCT для получения только неповторяющихся данных.

**WHERE**

Можно использовать ключевое слово WHERE в SELECT для указания условий в запросе

**GROUP BY**

Оператор GROUP BY часто используется с агрегатными функциями, такими как COUNT, MAX, MIN, SUM и AVG, для группировки выходных значений.

**HAVING**

Ключевое слово HAVING было добавлено в SQL по той причине, что WHERE не может использоваться для работы с агрегатными функциями.

**ORDER BY**

ORDER BY используется для сортировки результатов запроса по убыванию или возрастанию. ORDER BY отсортирует по возрастанию, если не будет указан способ сортировки ASC или DESC

**BETWEEN**

BETWEEN используется для выбора значений данных из определённого промежутка. Могут быть использованы числовые и текстовые значения, а также даты.

**LIKE**

Оператор LIKE используется в WHERE, чтобы задать шаблон поиска похожего значения.

Есть два свободных оператора, которые используются в LIKE:

% (ни одного, один или несколько символов);

\_ (один символ).

**IN**

С помощью IN можно указать несколько значений для оператора WHERE

**JOIN**

JOIN используется для связи двух или более таблиц с помощью общих атрибутов внутри них. На изображении ниже показаны различные способы объединения в SQL. Обратите внимание на разницу между левым внешним объединением и правым внешним объединением:

**VIEW**

VIEW — это виртуальная таблица SQL, созданная в результате выполнения выражения. Она содержит строки и столбцы и очень похожа на обычную SQL-таблицу. VIEW всегда показывает самую свежую информацию из базы данных.

**Концепция ролей**

Ролью в БД называется специальный объект, который используется для упрощения предоставления разрешений в БД. В отличие от серверных ролей, которые могут быть только встроенные роли БД могут быть так же созданы администратором. Указываются права доступа, имя владельца и имя роли.

При создании роли в графическом интерфейсе вы можете сразу назначить роль владельцем схемы в БД и назначить пользователей, которые будут обладать правами данной роли

**5. Преимущества СУБД InterBase. Назначение утилиты gbak, gfix, gsec в InterBase.**

**Сервер InterBase** — это кроссплатформенная СУБД, поддерживающая большинство операционных систем: Windows, Linux, Unix, Solaris, Mac OS и т.д.

InterBase обладает целым рядом преимуществ, выгодно отличающих его от остальных СУБД:

* 1. Учетные записи пользователей, групп и машин могут быть организованы в виде контейнеров каталога, называемых организационными подразделениями или просто подразделениями
  2. Поддерживается большое число объектов
  3. Возможности администрирования расширены за счет использования графических средств управления
  4. Служба тиражирования каталогов позволяет иметь несколько копий учетной информации, причем обновления этой информации могут выполняться в любой копии
  5. Хранение учетной информации в AD означает, что пользователи и группы представлены в виде объектов каталога.

Утилита **gbak**, входящая в комплект любого дистрибутива СУБД Firebird и InterBase, является средством создания резервных копий баз данных и восстановления баз данных из резервных копий.

Утилита **gfix** позволяет:

• Выполнять принудительную чистку (sweep) базы данных;

• Изменять интервал автоматической чистки;

• Закрывать базу данных для получения монопольного доступа, и затем снова открывать ее;

• Переводить базу в режимы "чтение/запись" или "только чтение";

• Переключаться между синхронным и асинхронным вводом (Forced Writes);

• Изменять диалект БД;

• Устанавливать размер кэша;

• Отыскивать повисшие транзакции и отменять или подтверждать их;

• Активировать или удалять теневые копии;

• Производить ремонт поврежденных баз данных.

**GSEC** - это утилита для работы с базой данных безопасности (содержащей информацию о пользователях СУБД). Она позволяет системному администратору (или другому привилегированному пользователю) управлять учетными записями пользователей для различных баз данных СУБД Firebird. Используя различные опции, можно добавлять, изменять или удалять учетные записи пользователей из базы данных безопасности.

**6. Преимущества создания резервных копий БД встроенными средствами СУБД. Теневые копии (shadow) БД. Основные рекомендации по ремонту БД.**

**Преимущества создания резервных копий БД встроенными средствами СУБД.**

* Возможность осуществлять резервное копирование БД параллельно с работой других пользователей. В начале копирования делается снимок системы, с которым и работает утилита копирования.
* Во время копирования считывается каждая запись из таблиц параллельно со сборкой мусора. Из копии пропадают устаревшие записи
* Оставшиеся данные переупаковываются, можно сказать дефрагментируются.
* Индексы пересчитываются
* Создание копии можно использовать для миграции БД на другой сервер

**Теневое** копирование (shadowing, аналог зеркалирования) является внутренним процессом, который поддерживает физическую **копию базы данных** в реальном времени. Всякий раз, когда изменения записываются в **базу данных**, **теневая копия** одновременно получает те же самые изменения. Присутствует возможность ручного перехода на теневую копию и продолжить системы.

**Рекомендации**

1. Отключение от БД всех пользователей
2. Сделать копию рабочей БД средствами файловой системы, все попытки восстановления делать с копией, а не с оригиналом
3. Провести полную проверку
4. Исправление разрушенных данных, пометить их недоступными
5. Вновь выполнить поиск разрушенных структур
6. Вновь выполнить резервное копирование
7. Если сработало – все хорошо. Если нет, то сделать еще одно копирование добавив параметр –g или –limbo
8. Восстановить резервную копию командой

**7. Определение каталога и службы каталогов. Назначение и функции служб каталога. Основные службы каталогов и стандарты, используемые в современных сетях.**

Каталог- информационный ресурс, используемый для хранения информации каком-либо объеме.

Служба каталогов – сетевые службы, которые идентифицирует все ресурсы сети и делает их доступными пользователем. Службы каталога отличаются от каталога тем, что они не только являются информационным ресурсом, но также предоставляет собой услугу, поиск и доставку необходимой пользователю информации.

**Назначения**

* Обеспечение защиты информации от вмешательства посторонних лиц в рамках, установленных администратором
* Распределение каталог среди других компьютеров в сети
* Проводить репликацию каталога, делая его доступным для большого числа пользователей и более защищенных данных от потери.
* Разделить каталог из нескольких частей, обеспечивая возможность хранения очень большого числа объектов

**Функции**

Централизация – уменьшение количества каталога в сети. Включение информации обо всех сетевых ресурсах в централизованный каталог создает единственную точку управления, что упрощает администрирование ресурсов и позволяет эффективнее делегировать задачи.

Масштабируемость – служба каталогов должна допускать рост сети, не создавая слишком больших издержек, - то есть она должна поддерживать какой-либо способ разбиения базы данных каталога на разделы, чтоб не утратить контроль над базой данных из-за чрезмерного разрастания и при этом сохранить преимущества централизации.

Стандартизация – служба каталогов должна предоставлять доступ к своей информации по открытым стандартам. Это гарантирует, что другие приложения смогут использовать ресурсы в службе каталогов, а не поддерживать собственные каталоги

Расширяемость – служба каталогов должна тем или иным способам позволять администраторам и приложениям расширять в соответствии с потребностями организации набора информации, хранимой в каталоге

Разделение физической сети - благодаря службе каталогов топология физической сети должна быть прозрачной для пользователей и администраторов. Ресурсы можно находить не зная, как и где они подключены к сети.

Безопасность – служба каталогов была бы крайне полезной злоумышленникам, так как хранит подробную информацию о данной организации. Поэтому служба каталогов должна поддерживать защищенные средства хранения, управления, выборки и публикации информации о сетевых ресурсах.

Active Directory – служба каталогов с Windows.

**Стандарт**

В течение 1980-х годов международной организацией International Telegraph and Telephone Consultative Committee разрабатывался общий стандарт для службы каталогов, в дальнейшем ставший называться [X.500](https://ru.wikipedia.org/wiki/X.500), частью которого является протокол доступа к каталогу [DAP](https://ru.wikipedia.org/w/index.php?title=Directory_Access_Protocol&action=edit&redlink=1) (Directory Access Protocol), используемый в современных службах каталогов в облегчённом варианте [LDAP](https://ru.wikipedia.org/wiki/LDAP) (Lightweight Directory Access Protocol) по причине первоначальной всеобъемлющей функциональности, оказавшейся неуместной для персональных компьютеров.

**8. Ключевые преимущества Active Directory. Область действия, пространство имен ActiveDirectory. Объект. Контейнер. Дерево.**

**Ключевые преимущества AD:**

* Централизованный каталог (единственная служба каталогов, реализованная в пределах предприятия, что упрощает сетевое администрирование; также каталог может использоваться с другими приложениями, например, Exchange Server 2000)
* Единая регистрация (после успешной идентификации пользователям будет предоставлен доступ ко всем сетевым разрешенным ресурсам, нет необходимости регистрироваться на различных серверах, доменах)
* Делегированное администрирование (возможность передачи прав, например, назначить определенной учетной записи пользователя административное право сбрасывать пароли в домене, но не создавать, удалять, изменять пользовательский объект)
* Интерфейс общего управления (использование консоли управления MMC)
* Интегрированная безопасность (аутентификация (процедура проверки подлинности входящего в систему объекта, предъявившего свой идентификатор) пользователей, назначается SID (идентификатор защиты) сравнивается с дискриптором защиты на ресурсе и устанавливается соответствие, уровень доступа)
* Масштабируемость (т.к. происходит рост организации (бизнес, слияние) нужно справляться с ее ростом, посредством добавления сервером или расширением размера доменной модели).

AD хранит информацию о сетевых ресурсах. Эти ресурсы (например, данные пользователей, описания принтеров, серверов, баз данных, групп, компьютеров и политик безопасности) называются объектами.

**Область действия** (scope) AD достаточно обширна: включает отдельные сетевые объекты (принтеры, файлы, имена пользователей), серверы и домены в отдельной глобальной сети. Может охватывать несколько объединенных сетей**.** AD может быть настроена на управление как отдельным компьютером, так и компьютерной сетью или группой сетей. AD, как и любая другая служба каталогов, является прежде всего пространством имен.

**Пространство имен** – это такая ограниченная область, в которой может быть распознано данное имя. Распознавание имени заключается в его сопоставлении с некоторым объектом или объемом информации, которому это имя соответствует. Файловая система Windows образует пространство имен, в котором имя файла может быть поставлено в соответствие конкретному файлу.AD образует пространство имен, в котором имя объекта в каталоге может быть поставлено в соответствие самому этому объекту.

**Объект** – непустой, именованный набор атрибутов, обозначающий нечто конкретное, например, пользователя, принтер или приложение.

Атрибуты содержат информацию, однозначно описывающую данный объект. Атрибуты пользователя могут включать имя пользователя, его фамилию и адрес электронной почты.

**Контейнер** он также имеет атрибуты и принадлежит пространству имен. Однако, в отличие от объекта, не обозначает ничего конкретного: он может содержать группу объектов или другие контейнеры.

**Термин "дерево"** используется в данном документе для описания иерархии объектов и контейнеров.

* Как правило, конечными элементами дерева являются объекты.
* В узлах (точках ветвления) дерева располагаются контейнеры.
* Дерево отражает взаимосвязь между объектами или указывает путь от одного объекта к другому.
* Простой каталог представляет собой контейнер.
* Компьютерная сеть или домен тоже являются контейнерами.
* Непрерывным поддеревом называют любую непрерывную часть дерева, включающую все элементы каждого входящего в нее контейнера.

**9. Домен. Доменное дерево. Лес. Организационные единицы. Сайт, узел.**

**Домен** – это единая область, в пределах которой обеспечивается безопасность данных в компьютерной сети под управлением ОС Windows.

AD состоит из одного или нескольких доменов. Применительно к отдельной рабочей станции доменом является сама станция. Границы одного домена могут охватывать более чем одно физическое устройство. Каждый домен может иметь свои правила защиты информации и правила взаимодействия с другими доменами. Если несколько доменов связаны друг с другом доверительными отношениями и имеют единую логическую структуру, конфигурацию и глобальный каталог, то говорят о **дереве доменов.**

Дерево доменов состоит из нескольких доменов, которые имеют общую логическую структуру и конфигурацию и образуют непрерывное пространство имен. Домены в дереве связаны между собой доверительными отношениями. AD является множеством, которому принадлежат одно или несколько деревьев доменов.

Дерево доменов графически можно представить двумя способами:

* Представление доменного дерева через доверительные отношения между доменами.
* Представление доменного дерева через пространство имен доменного дерева.

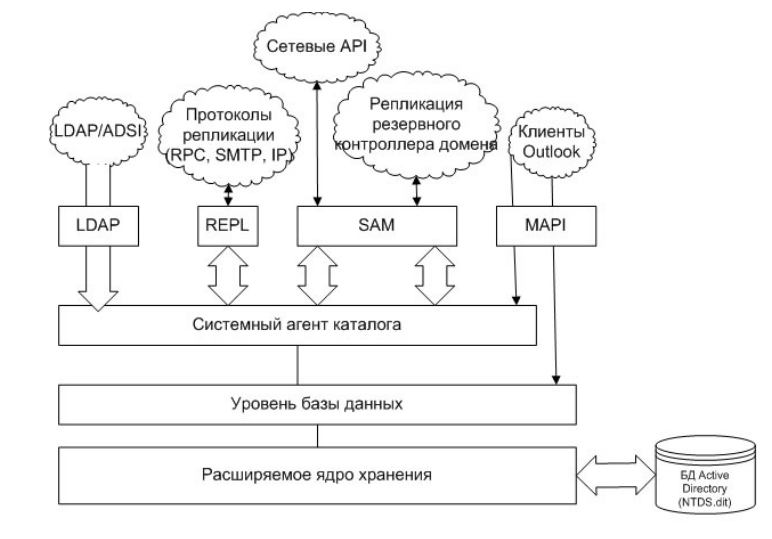
Несколько доменных деревьев могут быть объединены в **лес** – одно или несколько деревьев, которые не образуют непрерывного пространства имен.Все деревья одного леса имеют общие логическую структуру, конфигурацию и глобальный каталог, поддерживают друг с другом транзитивные иерархические доверительные отношения, устанавливаемые на основе протокола Kerberos.В отличие от дерева, лес может не иметь какого-то определенного имени.

**Организационные единицы** (Organizational Units, OU) или организационные подразделения (ОП) позволяют разделять домен на зоны административного управления, т.е. создавать единицы административного управления внутри домена.

**Узлом (сайтом)** называется такой элемент сети, который содержит серверы AD. Узел обычно определяется как одна или несколько подсетей, поддерживающих протокол TCP/IP и характеризующихся хорошим качеством связи, которое подразумевает высокую надежность и скорость передачи данных. Определение узла как совокупности подсетей позволяет администратору быстро и без больших затрат настроить топологию доступа и репликации в AD и полнее использовать достоинства физического расположения устройств в сети.

**10. Функциональная структура Active Directory. Физическая структура Active Directory. Логическая структура Active Directory.**

**Функциональную структуру** AD можно представить в виде многоуровневой архитектуры, в которой уровни являются процессами, предоставляющими клиентским приложениям доступ к службе каталога. AD состоит из трех уровней служб и нескольких интерфейсов и протоколов, совместно работающих для предоставления доступа к службе каталога. Три уровня служб охватывают различные типы информации, необходимой для поиска записей в базе данных (БД) каталога. Выше уровней служб в этой архитектуре находятся протоколы и API-интерфейсы, осуществляющие связь между клиентами и службой каталога.



**Физическая структура сети** с AD довольно проста по сравнению с ее логической структурой, потому что физические компоненты AD – это узлы (сайты) и контроллеры домена (компьютер-сервер, управляющий доменом и хранящий полную копию той части каталога, которая относится к их домену). Необходимо четко представлять, что компонентами логической структуры AD являются домены, тогда как компонентами физической структуры являются сайты.

Деление на сайты не зависит от доменной (логической) структуры, т.е.:

* в сайте может быть один домен (либо только его часть) или несколько доменов;
* в домене (или даже в организационном подразделении) может быть несколько сайтов.

Сайты содержат объекты только двух типов:

* контроллеры доменов в границах сайта;
* связи сайта (site links), сконфигурированные для соединения данного сайта с остальными.

Компонентами логической структуры AD являются домены, тогда как компонентами физической структуры являются сайты.

• Домен – объединение серверов и других сетевых ресурсов, собранных под одним именем.

• Сайт – комбинация одной или более IP-подсетей, которые должны быть соединены высокоскоростным каналом связи.

В AD ресурсы организованы в **логическую структуру**, отражающую структуру организации, что позволяет находить ресурс по его имени, а не по физическому расположению. Благодаря логическому объединению ресурсов в AD физическая структура сети не важна для пользователей. Логическая структура AD является моделью службы каталога, которая определяет каждого участника безопасности на предприятии, а также организацию этих участников.

Логические компоненты AD:

* Объекты – ресурсы хранятся в виде объектов.

-Классы объектов.

-Схема Active Directory.

* Домены – базовая организационная структура.
* Деревья – несколько доменов объединяются в иерархическую структуру.
* Леса – группа из нескольких деревьев домена.
* Организационные единицы – позволяют делить домен на зоны и делегировать права на них.

Логическая структура AD не базируется на физическом местонахождении серверов или сетевых соединениях в пределах домена, что позволяет структурировать домены, отталкиваясь не от требований физической сети, а от административных и организационных требований.

**11. Какие сведения собираются на этапе предпроектного исследования для проектирования службы каталогов. Типовой план проектирования структуры ActiveDirectory. Какие объекты содержит БД ActiveDirectory.**

На **этапе предпроектного исследования** собираются основные сведения по существующей инфраструктуре в компании.

Для планирования структуры Active Directory – информация о доменной структуре и ее типе, структуре групп пользователей и распределении их по доменам, количестве существующих контроллеров доменов внутри каждого домена.

Определение

– существующих доверительных отношений между доменами,

– односторонних и двухсторонних доверительных отношений, и доменов, которые не должны включаться в леса Active Directory,

– пространства имен DNS, существующих в организации,

– и перечня существующих доменных имен организации, зарегистрированных в сети Интернет.

Для планирования сайтов Active Directory – информация о существующей структуре сайтов, о топологии сети, о каналах связи и их пропускной способности.

Для планирования переноса текущей структуры сетевых сервисов на платформу Active Directory – информация о топологии используемых сетевых сервисов DHCP, WINS, DNS.

Для обеспечения возможности резервного восстановления данных во время миграции – схема резервного копирования информации.

Для определения возможной расширяемости решения – исследование возможных вариантов изменения схемы при росте организации или ее реорганизации, определение области Active Directory (исследование подразделений, включая удаленные филиалы организации, необходимых для включения в Active Directory).

Для планирования миграции приложений, использующих Active Directory – список приложений, связанных с Active Directory, и возможных ограничений, накладываемых ими на структуру Active Directory, определение механизмов идентификации пользователей.

Проектирование структуры Active Directory начинается с компонентов высшего уровня, а затем проектируются компоненты низших уровней. Это означает, что:

• первый шаг состоит в создании проекта леса,

• затем следует проект доменов,

• проект DNS

• и, наконец, проект организационной единицы (OU).

**План проектирования:**

* 1. Планирование структуры лесов:

– Определение типовых лесов для основных типов региональных представительств.

– Определение основных типов доверительных отношений между разными лесами.

– Разработка политики контроля изменений леса.

– Политика изменения схемы.

– Политика изменения конфигурации.

– Разработка структуры DNS для типовых лесов.

2. Планирование доменов для каждого леса:

– Реструктуризация существующих доменов на домены в зависимости от административных потребностей.

– Разбиение на домены в зависимости от физического месторасположения для оптимизации трафика репликации и запросов.

– Выбор корневого домена для каждого леса.

– Оптимизация аутентификации укороченными доверительными отношениями.

3. Планирование использования сайтов для каждого леса:

– Определение сайтов на основании физической топологии сети.

– Создание связей между сайтами.

– План размещения серверов глобального каталога в сайтах.

– План размещения серверов DNS в сайтах.

4. Планирование структуры организационных единиц для каждого домена:

– Планирование реорганизации существующих доменов в организационных единицах.

– Планирование организационных единиц для делегирования (передача части функций руководителя другим сотрудникам) административных полномочий.

– Планирование организационных единиц для скрытия объектов.

– Создание организационных единиц для применения групповых политик.

5. Планирование реорганизации существующих доменов и их перевод на новую платформу AD:

– Планирование перемещения пользователей, компьютеров и групп.

– Планирование модификации или удаления из структуры реструктуризируемых доменов.

– Планирование изменения существующих доверительных отношений.

– Планирование клонирования объектов безопасности.

6. Тестирование внедряемых решений и установка стенда:

– Определение возможностей и целей тестирования.

– Разработка сценариев тестирования:

цель тестирования;

тестируемые возможности и функции;

требования к оборудованию, программному обеспечению и его конфигурации;

описание проведения тестирования;

ожидаемые результаты или критерии успеха теста; график тестирования.

**БД Active Directory содержит** следующие структурные объекты:

• Разделы (сегменты) называются контекстами именования (NC – Naming Contexts). Существует: раздел домена каталога, раздел конфигурации каталога, раздел схемы каталога, раздел глобального каталога, разделы приложений каталога.

• Домены. Домен служит в качестве административной границы, определяет и границу политик безопасности. Каждый домен имеет, хотя бы один (оптимально более), контроллер домена (компьютер-сервер, управляет доменом и содержит копию той части каталога, которая относится к их домену). Домены AD организованы в иерархическом порядке. Первый на предприятии становится корневым доменом леса, называется корневым доменом или доменом леса.

• Деревья доменов. Домены, которые создаются в инфраструктуре AD после создания корневого домена, могут использовать существующее пространство имен AD совместно или иметь отдельное пространство имен. Чтобы выделить отдельное пространство имен для нового домена, нужно создать новое дерево домена.

• Леса. Лес определяет границу безопасности для предприятия, являясь общим для всех контроллеров домена в лесу. Все домены и доменные деревья существуют в пределах одного или несколько лесов AD.

• Сайты. Сайт представляет область сети, где все контроллеры домена связаны быстрым, недорогим и надежным сетевым подключением. Независимость логических компонентов от сетевой инфраструктуры возникает вследствие использования сайтов в AD: они обеспечивают соединение между логическими компонентами AD и физической сетевой инфраструктурой.

• Организационные единицы. Организационные единицы предназначены для того, чтобы облегчить управление службой AD. Они служат для создания иерархической структуры в пределах домена и используются, чтобы сделать более эффективным управление единственным доменом (вместо управления несколькими доменами AD).

**12. Доверительные отношения между доменами. Типы доверительных отношений. Контроллеры доменов, функции контроллера домена, их роли.**

Если несколько доменов связаны друг с другом **доверительными отношениями** и имеют единую логическую структуру, конфигурацию и глобальный каталог, то говорят о дереве доменов.

Доверительные отношения – это механизм, который позволяет объектам в одном домене [доверяемом (trusted domain)] обращаться к ресурсам в другом [доверяющем (trusting domain)], так как домены разграничивают зоны безопасности.

Windows Server 2003 поддерживает шесть **типов доверительных отношений**:

• Доверие к родительскому и дочернему доменам. AD автоматически выстраивает транзитивные двусторонние доверительные отношения между родительскими и дочерними доменами в дереве доменов. При создании дочернего домена доверительные отношения автоматически формируются между дочерним доменом и его родителем. Эти отношения двусторонние. Доверие также является транзитивным, т.е. контроллеры доверяемого домена пересылают запросы на аутентификацию (проверка подлинности входящего в систему субъекта по ID) контроллерам доверяющих доменов.

• Доверие к корневому домену дерева. Двусторонние транзитивные доверительные отношения автоматически создаются и между корневыми доменами деревьев в одном лесу. Это резко упрощает управление доменами по сравнению с тем, что было в версиях Windows, предшествовавших Windows 2000. Больше не нужно конфигурировать отдельные односторонние доверительные отношения между доменами.

• Доверие к внешнему домену. Внешнее доверие используется, когда нужно создать доверительные отношения между доменом Windows Server 2003 и доменом Windows NT 4.0. Поскольку ограниченные домены (down-level domains), которые не поддерживают AD, не могут участвовать в двусторонних транзитивных доверительных отношениях, следует использовать внешнее доверие, которое является односторонним.

• Доверие к сокращению. Способ создания прямых доверительных отношений между двумя доменами, которые могут быть уже связаны цепочкой транзитивных доверий, но нуждаются в более оперативном реагировании на запросы друг от друга.

• Доверие к сфере. Служит для подключения домена Windows Server 2003 к сфере Kerberos, которая не поддерживает Windows и использует протокол защиты Kerberos V5. Доверие к сфере может быть транзитивным или нетранзитивным, одно- или двусторонним.

• Доверие к лесу. Упрощает управление несколькими лесами и обеспечивает более эффективное защищенное взаимодействие между ними. Этот тип доверия позволяет обращаться к ресурсам в другом лесу по той же идентификации пользователя (user IDentification, ID), что и в его собственном лесу.

**Контроллер домена** – компьютер-сервер, управляющий доменом и хранящий реплику каталога домена (локальную БД домена). Поскольку в домене может быть несколько контроллеров домена, все они хранят полную копию той части каталога, которая относится к их домену.

**Функции контроллеров доменов:**

• Каждый контроллер домена хранит полную копию всей информации AD, относящейся к его домену, а также управляет изменениями этой информации и реплицирует их на остальные контроллеры того же домена.

• Все контроллеры в домене автоматически реплицируют (процесс синхронизации изменений, сделанных на одном контроллере домена с другими контроллерами домена) между собой все объекты в домене. Какие-либо изменения, вносимые в AD, на самом деле производятся на одном из контроллеров домена. Затем этот контроллер домена реплицирует изменения на остальные контроллеры в пределах своего домена. Задавая частоту репликации и количество данных, которое Windows будет передавать при каждой репликации, можно регулировать сетевой трафик между контроллерами домена.

• Важные обновления, например, отключение учетной записи пользователя, контроллеры домена реплицируют немедленно.

• AD использует репликацию с несколькими хозяевами (multimaster replication), в котором ни один из контроллеров домена не является главным. Все контроллеры равноправны, и каждый из них содержит копию бд каталога, в которую разрешается вносить изменения. В короткие периоды времени информация в этих копиях может отличаться до тех пор, пока все контроллеры не синхронизируются друг с другом.

• Наличие в домене нескольких контроллеров обеспечивает отказоустойчивость. Если один из контроллеров домена недоступен, другой будет выполнять все необходимые операции, например, записывать изменения в Active Directory.

• Контроллеры домена управляют взаимодействием пользователей и домена, например, находят объекты Active Directory и распознают попытки входа в сеть.

**Роли контроллера домена:**

Две роли хозяина операций, которые могут быть назначены единственному контроллеру домена в лесу (роли, действующие в границах леса):

• Хозяин схемы (Schema Master). Первый контроллер домена в лесу принимает роль хозяина схемы и отвечает за поддержку и распространение схемы на остальную часть леса, поддерживает список всех возможных классов объектов и атрибутов, определяющих объекты, которые находятся в AD. Если схему нужно обновлять или изменять, наличие Schema Master обязательно.

• Хозяин именования доменов (Domain Naming Master). Протоколирует добавление и удаление доменов в лесу и жизненно необходим для поддержания целостности доменов. Domain Naming Master запрашивается при добавлении к лесу новых доменов. Если Domain Naming Master недоступен, то добавление новых доменов невозможно; однако при необходимости эта роль может быть передана другому контроллеру.

Три роли хозяина операций, которые могут быть назначены одному из контроллеров в каждом домене (общедоменные роли).

• Хозяин RID (Relative Identifier (RID) Master). Отвечает за выделение диапазонов относительных идентификаторов (RID) всем контроллерам в домене.

SID (идентификатор безопасности) в Windows Server 2003 состоит из двух частей. Первая часть – общая для всех объектов в домене; для создания уникального SID к этой части добавляется уникальный RID. Вместе они уникально идентифицируют объект и указывают, где он был создан.

• Эмулятор основного контроллера домена (Primary Domain Controller (PDC) Emulator). Отвечает за эмуляцию Windows NT 4.0 PDC для клиентских машин, которые еще не переведены на Windows 2000, Windows Server 2003 или Windows XP и на которых не установлен клиент службы каталогов. Одна из основных задач эмулятора PDC - регистрировать устаревшие клиенты. Кроме того, к эмулятору PDC происходит обращение, если аутентификация клиента оказалась неудачной. Это дает возможность эмулятору PDC проверять недавно измененные пароли для устаревших клиентов в домене, прежде чем отклонять запрос на вход.

• Хозяин инфраструктуры (Infrastructure Master, IM). Регистрирует изменения, вносимые в контролируемые объекты в домене. Обо всех изменениях сначала сообщается IM, и лишь потом они реплицируются на другие контроллеры домена. IM обрабатывает информацию о группах и членстве в них для всех объектов в домене, передают информацию об изменениях, внесенных в объекты, в другие домены.

**13. ИТ решения для индивидуальной и коллективной работы пользователей корпоративных информационных систем (интегрированные средства коммуникаций, рабочие области коллективной деятельности, мгновенный доступ к информации и людям, автоматизация бизнес-процессов). Решения Microsoft Windows.**

Поддержка индивидуальной и коллективной работы пользователей корпоративных ИС может быть реализована на базе **решений**:

1. Интегрированные средства коммуникаций. Так как ИТ-инфраструктура предприятия должна обеспечивать взаимодействие всех средств коммуникаций (городские и сотовые телефоны, смартфоны, КПК (карманные ПК), ПК, ноутбуки и Internet-киоски), которые используют сотрудники предприятий для доступа к информации, существуют решения Microsoft, которые упрощают и интегрируют разнообразные средства коммуникаций, доступные группам и индивидуальным сотрудникам. Электронная почта, мгновенный обмен сообщениями, голосовая почта, телефоны, мобильные устройства и средства проведения конференций через Internet объединяются унифицированным ПО, его функции должны быть доступны независимо от места нахождения пользователейили типа сетевого соединения.

То есть такое интеллектуальное ПО, которое управляет коммуникациями, с учетом возможностей линий связи, в реальном масштабе времени. Данное ПО способствует созданию эффективных коммуникаций как внутри предприятия, так и с партнерами, поставщиками и клиентами.

2. Рабочие области коллективной деятельности. Для поддержки коллективной работы Microsoft предлагает использовать службу Windows SharePoint Services, которая устанавливается в Microsoft Windows Server 2003. Данная служба:

• предоставляет надежные и простые в использовании рабочие области для групп,

• легко интегрируется с Microsoft Windows Server, позволяя ИТ-службе создавать рабочие области коллективной работы. Эти области облегчают проведение совещаний, управление проектами, создание документов и др. Windows SharePoint Services можно интегрировать с корпоративными бизнес-приложениями и, следовательно, получать к ним доступ посредством привычного пользователю интерфейса.

3. Мгновенный доступ к информации и людям. Так как корпоративным пользователям требуются эффективные средства поиска информации во множестве источников, Microsoft предлагает использовать портальные технологии и управление контентом. С помощью функциональности MySites в SharePoint пользователи могут создавать свои сайты под личные задачи. Такой сайт является единой точкой доступа к документам пользователя, новостям, электронной почте и другим приложениям.

4. Автоматизация бизнес-процессов. При автоматизации внутренних бизнес-процессов предприятия появляется необходимость исключения бумажного документооборота из информационных потоков. Решения Microsoft: используют привычные программы, такие как Microsoft Windows Server, для обращения к корпоративной информации и приложениям. При интеграции ERP-систем с программами Microsoft Windows Server сотрудники предприятия могут обращаться к бизнес-приложениям прямо из Microsoft Windows Server. Поддержка XML в Microsoft Windows Server предоставляет большие возможности по формированию индивидуальных схем информационных потоков и позволяет применять гибкие средства управления процессами на основе документов.

**14. Платформы для построения корпоративных порталов. Расписать (Microsoft, IBM, Oracle).**

Рынок платформ для построения корпоративных порталов в настоящее время является растущим. Сам «корпоративный портал» представляет собой специальную среду, объединяющую классические инструменты, такие как работа с документами, задачи, календари, а также новые инструменты, среди которых: корпоративная социальная сеть, блоги и настоящие интерактивные коммуникации. Основной тренд сегодня – это повышение эффективности бизнеса за счет повышения отдачи и создания продуктивной среды работы для сотрудников, а также раскрытие потенциала людей.

Платформы:

1. Microsoft Share Point (Share Point, или Microsoft Share Point Products and Technologies) представляет собой рабочую платформу для создания сайтов и корпоративных порталов, интегрированные функции которой предоставляют широкие возможности коммуникации между людьми.

Существует несколько версий: Share Point 2010, 2013, 2016.

Службы 2016 года:

• Share Point Online – облачная служба, размещенная в Microsoft и предназначенная для предприятий любых размеров. Вместо локальной установки и развертывания Share Point Server любая организация может просто подписаться на план Office 365 или отдельную службу Share Point Online, чтобы сотрудники могли создавать сайты для доступа к документам и данным совместно с коллегами, партнерами и клиентами.

• Share Point Server – организации могут развернуть данную систему и управлять ею локально. К ее функциям относятся управление корпоративным контентом, бизнес-аналитика, поиск в корпоративной среде, личные сайты и канал новостей, а также все функции SharePoint Foundation

• Share Point Foundation – технология, которая использовалась в качестве основы для всех сайтов Share Point, больше не предоставляется отдельно для выпуска Share Point 2016. Share Point Foundation (прежнее название – Windows Share Point Services) предлагается бесплатно для локального развертывания. Ее можно использовать для создания сайтов различного типа, совместной работы над веб-страницами, документами, списками, календарями и данными, используется для:

– координирования расписаний;

– организации документов;

– участия в обсуждениях с помощью рабочих областей группы, блогов, вики-сайтов и библиотек документов.

• Share Point Designer 2013 – бесплатная программа (последний выпуск в 2013 г.) для создания функциональных решений с поддержкой рабочих процессов и редактирования внешних типов контента для работы с внешними данными.

• Синхронизация One Drive для бизнеса – программа для настольных систем, с помощью которой можно синхронизировать документы с сайта группы или из One Drive для бизнеса с компьютером для работы в автономном режиме.

2. IBM Web Sphere Portal. Передовые портальные решения IBM способствуют повышению удобства работы с веб-приложениями, позволяют защитить и сохранить инвестиции в существующие системы, повышают производительность и создают возможности для дальнейшего роста компании.

Три портальных решения: IBM WebSphere Portal Server, Enable, Extend.

Web Sphere Portal включает в себя набор портальных технологий, которые позволяют разрабатывать и поддерживать порталы высокого класса для взаимодействия компаний с заказчиками (В2С), сотрудниками (В2Е) и другими компаниями (В2В). Этот продукт ускоряет окупаемость капиталовложений за счет следующих преимуществ:

• Максимальное удобство веб-интерфейса – можно воспользоваться всеми достижениями технологии Web 2.0 для реализации современных бизнес-моделей.

• Расширение бизнес-активов – позволяет создавать и внедрять собственные, ориентированные на потребности рынка решения на основе новых и существующих бизнес-активов, используя их повторно и меняя их назначение при необходимости.

• Повышение эффективности работы – проверенный, надежный, масштабируемый и высокопроизводительный фундамент для корпоративных приложений, процессов и транзакций.

• Обеспечение развития компании – позволяет «подключать» рассчитанные на конкретные бизнес-потребности инструменты, которые ускорят получение отдачи от решения и позволят использовать новые коммерческие возможности рынка, одновременно снижая затраты на развертывание порталов.

• Новая версия продукта Web Sphere Portlet Factory содержит еще более простые шаблоны, использует новые возможности технологии Web 2.0 и поддерживает средства визуальной разработки настольных веб-приложений.

3. Oracle Portal. Эффективно решает все основные задачи, предоставляя корпоративным пользователям интуитивный доступ к необходимым данным и сервисам, а также позволяет бесшовно интегрировать корпоративные приложения.

Основные составляющие:

• средства построения порталов на основе портальных компонентов;

• библиотеки готовых компонентов;

• средства управления содержанием с поддержкой коллективной работы;

• средства для сквозного полнотекстового поиска информации на портале, в интернете (Интранете), в бд и в хранилищах неструктурированной информации;

• средства интеграции внешних систем, приложений, различных видов источников данных;

• средства для анализа бизнес-информации (Business Intelligence);

• средства однократной регистрации и управления информационной безопасностью;

• средства поддержки мобильного беспроводного доступа;

• интегрированное средство разработки веб-приложений.

Oracle Portal состоит из следующих ключевых компонентов:

• Application Development Framework (ADF) – технология разработки приложений;

• Web Center Portal: Framework – функциональное развитие технологии ADF для специфичных функциональных требований портала;

• Web Center Portal: Spaces – готовый к использованию, преднастроенный инструмент создания порталов совместной работы;

• Web Center Portal: Services – набор социальных инструментов портала;

• Composer – инструмент создания и модификации страниц, структуры портала без программирования;

• Discussion Server – реализация форумов;

• Analytics – механизм сбора статистики работы пользователей с объектами портала;

• Activity Graph – инструмент визуализации активностей пользователей;

• Personalization Server – набор функций для выполнения задач кастомизации портального интерфейса как пользователем, так и ГГ-специалистами;

• Portals – механизм создания отдельных порталов для пользователей и групп пользователей, «портал в портале»;

• Composite Applications – среда исполнения и интеграции композитных приложений.

Современные корпоративные сайты и порталы, в частности порталы знаний, являются основой корпоративных информационных систем. Средства их разработки постоянно развиваются, видоизменяются, адаптируясь к развитию веб-технологий, а также пожеланиям пользователей. Жизненный цикл средств для построения сайтов и порталов составляет примерно 3-5 лет.

**15. Брандмауэры. Определение, общая информация, схема работы, Demilitarized Zone.**

Общепринятого определения термина **«брандмауэр»** не существует.

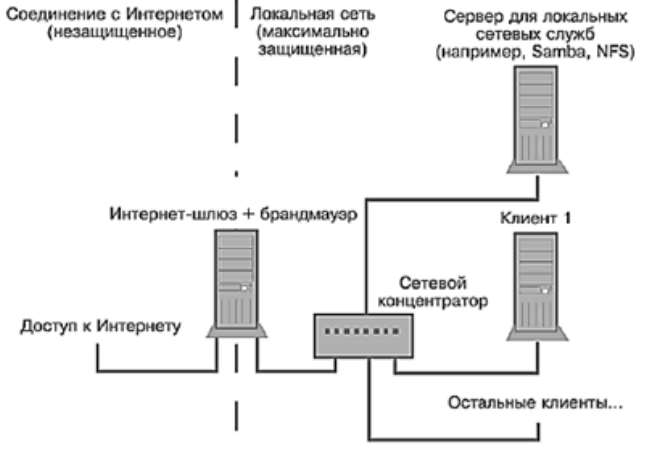
Функции брандмауэра могут выполняться оборудованием: в таком случае под брандмауэром обычно понимается компьютер, стоящий на стыке локальной сети и Интернета.

Нередко брандмауэром может быть и программный пакет, установленный на компьютере и при условии правильной конфигурации повышающий безопасность компьютера. Во многих дистрибутивах содержатся многофункциональные инструменты, предназначенные для настройки конфигурации брандмауэра.

Под брандмауэром будем понимать совокупность методов, повышающих надежность обмена информацией, проходящей по TCP/IP через фильтр пакетов.

Такой фильтр анализирует все сетевые пакеты, приходящие на компьютер, а также пакеты, которые уходят с компьютера в сеть. В зависимости от того, все ли правила соблюдаются, пакеты могут быть пропущены или заблокированы.

**Схема работы для локальных сетей:** в фирменной локальной сети за выход в Интернет и обеспечение безопасности часто отвечает отдельный компьютер. Все остальные сетевые службы работают на других компьютерах. Эта концепция изображена на рисунке:



В очень небольших сетях функции брандмауэра и сетевого сервера может выполнять один компьютер. Но такой подход не является оптимальным, так как на этом компьютере придется запустить и эксплуатировать множество сетевых служб, которые могут быть использованы для нанесения вреда сети.

В очень больших сетях часто бывает два брандмауэра. Первый служит лишь для обеспечения базовой безопасности, но пропускает такие интернет-протоколы, как HTTP или FTP. Сетевое пространство в таком случае называется **демилитаризованной зоной** (Demilitarized Zone, DMZ), то есть, внутри сети существуют ограниченные меры безопасности. В этой зоне располагается веб-сервер, а также другие сетевые серверы, которые должны быть общедоступны (то есть могут быть найдены через Интернет).

DMZ отделяется от оставшейся части локальной сети вторым брандмауэром. Уже за ним располагаются все другие службы, отвечающие за работу локальной сети и абсолютно недоступные извне. Однако конфигурация многоступенчатого брандмауэра – очень обширная тема. Руководства по конфигурации таких барьеров даются в специальной литературе.

**16. Методы виртуализации операционных системы. Характеристика метода, примеры фирм и программ.**

**Полная виртуализация (виртуальные машины, эмуляция)**

В данном случае программа имитирует работу виртуального аппаратного обеспечения, то есть компьютера, состоящего из процессора, ОЗУ, жесткого диска, сетевой карты и т. д.

Программы/фирмы: VMware, QEMU, Parallels, VirtualBox, Microsoft Virtual PC.

**Паравиртуализация**

В данном случае хозяин также же предоставляет виртуальные машины, на которых выполняются программы гостей. Отличие от полной виртуализации состоит в том, что гостевую операционную систему для виртуализации требуется модифицировать, после чего эта система напрямую сообщается с VMM.

Программы/фирмы: Xen, UML (Linux в пользовательском режиме).

**(Пара)виртуализация с поддержкой аппаратного обеспечения**

Современные процессоры производства Intel и AMD содержат аппаратные функции, предназначенные для упрощения процессов виртуализации.

Программы/фирмы: KVM, Xen.

**Виртуализация на уровне операционной системы (контейнеры)**

При использовании данного метода настоящие виртуальные машины не применяются. Вместо этого при таком подходе машины применяют общее ядро и фрагменты файловой системы хозяина.

Программы/фирмы: OpenVZ, Virtuozzo, Linux-VServer.

**17. Модель управления для администрирования сетевых систем FCAPS.**

Согласно модели FCAPS, все аспекты управления сетью могут быть описаны с помощью пяти областей управления:

• (F) Fault Management (управление отказами) – обнаружение отказов в устройствах сети, сопоставление аварийной информации от различных устройств, локализация отказов и инициирование корректирующих действий.

• (С) Configuration Management (управление конфигурированием) – возможность отслеживания изменений, конфигурирования, передачи и установки ПО на всех устройствах сети.

• (A) Accounting Management (управление учетом) – возможность сбора и передачи учетной информации для генерации отчетов об использовании сетевых ресурсов.

• (Р) Performance Management (управление производительностью) – непрерывный источник информации для мониторинга показателей работы сети (QoS, ToS) и распределения сетевых ресурсов.

• (S) Security Management (управление безопасностью) – возможность управления доступом к сетевым ресурсам.

В первой группе рассматривается управление администрацией системы отказами и соответствующие действия по поиску и диагностике ошибок системы, приводящих к отказам или ухудшению производительности системы.

Включает выявление, определение и устранение последствий сбоев и отказов в работе сети. На этом уровне выполняется не только регистрация сообщений об ошибках, но и их фильтрация, маршрутизация и анализ на основе знаний и опыта администратора системы.

Фильтрация позволяет выделить только важные сообщения из весьма интенсивного потока сообщений об ошибках, который обычно наблюдается в большой сети.

Маршрутизация обеспечивает их доставку нужному элементу системы управления, а корреляционный анализ позволяет найти причину, породившую поток взаимосвязанных сообщений.

Например, обрыв кабеля может быть причиной большого количества сообщений о недоступности сетей и серверов

В FCAPS идентифицировано 12 задач управления АС как необходимых для успешной работы по управлению отказами и поиску ошибок:

1) определение ошибки;

2) коррекция ошибки;

3) изоляция ошибки;

4) восстановление после ошибки;

5) поддержка тревожных сигналов (alarms);

6) фильтрация тревожных сигналов;

7) генерация тревожных сигналов;

8) проблема объяснения ошибки (корреляция);

9) проведение диагностических тестов;

10) ведение журнала ошибок;

11) сбор статистики ошибок;

12) сопровождение ошибок.

АС должен понимать, что управляющая система помогает ему, а не думает за него. Помимо управляющей системы, а также в ситуации, когда она не используется вовсе, АС должен пользоваться моделью поиска ошибок, которую рекомендуют обычно разработчики операционных систем.

**18. Системы управления MS (Management System), NMS (Network Management System). Схема работы NMS.**

Устранение ошибок в системе может быть автоматическим и полуавтоматическим.

• При автоматическом устранении ошибок ИС непосредственно управляет оборудованием или программными комплексами и обходит отказавший элемент за счет резервных каналов или специальных технологий, например, протоколов.

• В полуавтоматическом режиме основные решения и действия по устранению неисправности выполняют службы АС, а специализированная **система управления MS** (Management System) только помогает в организации этого процесса, например, оформляет квитанции на выполнение работ и отслеживает их поэтапное выполнение, а не устраняет аппаратные или кабельные проблемы. MS является специализированным ПО, которое ведет журнал ошибок, собирает статистику, фиксирует конфигурации средств системы, опознает тревожные ситуации.

Для управления только сетевыми системами используют **NMS** (Network Management System). Обычно при реализации своих функций NMS использует протокол SNMP (интернет-протокол для управления устройствами в IP-сетях на основе архитектур TCP/UD).

**Схема работы NMS:**

• SYSLOG – это сервер, который собирает все журналы (логи) системы, например, журнал ошибок, журнал сообщений.

• На коммутаторе работает программный продукт – SNMP-агент, который посылает информацию о своей деятельности по протоколу SNMP специальному серверу NMS, где работает другой программный продукт – SNMP-менеджер.

• Агенты SNMP могут работать и на файл-сервере (FS) и на сервере БД (DBS).

• Информация собирается менеджером в БД MIB для дальнейшего анализа и соответствующих действий администратора системы и NMS.

**19. Базовая модель поиска ошибок при администрировании ИС.**

Базовая модель поиска ошибок предусматривает последовательное выполнение администратором системы следующих действий:

1. Убедиться в том, что ошибки действительно есть. После сообщения пользователя о некорректной работе ИС надо убедиться в том, что этот пользователь выполняет все процедуры корректно и правильно оценивает работу ИС. Например, некая операция действительно занимает много времени, а пользователь считает, что ИС медленно работает.

2. Провести инвентаризацию. То есть, необходимо выяснить, все ли части ИС на месте: все кабели существуют, все части ИС взаимодействуют и правильно соединены. При этом NMS может помочь провести автоматический опрос параметров работы оборудования и ПО, дать план системы. У АС должна быть исполнительная документация по ИС с картой сети и списками всех параметров загрузки серверов, рабочих станций, коммутационного оборудования (worksheet). Нужно убедиться в том, что «все на месте» и соответствует документации.

3. Сделать копии ИС (backup). Причем желательно это делать «быстрыми средствами» (например, не утилитой копирования СУБД, а утилитами ОС «том в том» или «диск в диск»).

4. Сделать перезагрузку всех компонент ИС (restart). Есть два режима перезагрузки: холодный режим (с отключением питания) и горячий режим (без отключения питания). При холодном рестарте заново загружается все ПО оборудования, все драйверы, все процессы ОС и СУБД, заново инициализируется память серверов. Поэтому при ошибочных ситуациях надо использовать холодный рестарт. Однако если есть ошибки оборудования, то оно после этого может вообще не загрузиться. Перед перезагрузкой нужна не забыть завершить работу всех процессов различных ОС и СУБД (обычно команды типа Down или Shutdown).

5. После перезагрузки необходимо упростить работу ИС, например, завершить работу всех резидентных программ, не обязательных для работы в простейшем варианте ИС.

6. Если система загрузилась, нужно проверить права и привилегии работающих пользователей (например, одно приложение запускается и работает нормально с данными правами пользователя, а другое нет).

7. Надо убедиться, что версии ПО являются текущими. Следует работать не на последней версии продуктов, а на стабильной, хорошо отлаженной. Нужно убедиться в том, что никто из пользователей не поставил себе никаких обновлений ПО. Хотя при правильных действиях АС и NMS такой возможности у пользователя не должно быть.

8. Только после всех перечисленных действий надо собирать информацию об ошибке. Для этого следует проанализировать журналы ИС (логи). Выявить симптомы проблемы, а также тех, кто был ею затронут, проанализировать использование процессов во время возникновения ошибки, изменения, произошедшие в системе, после которых появились сообщения об ошибке в журналах.

9. Необходимо разработать план по изоляции ошибки. Для этого строятся гипотезы о причинах ошибки в ИС. Это могут быть ошибки каналов связи (80% всех ошибок), аппаратные ошибки, ошибки системного ПО, прикладного ПО. Всегда следует учитывать, что тираж аппаратных средств больше, чем тираж программных продуктов. Например, процессоров Intel выпускается больше, чем установок какой-либо одной ОС, поэтому аппаратных ошибок будет меньше, чем программных. Аналогично тираж системного ПО больше, чем тираж прикладного ПО, поэтому в первом меньше ошибок, чем в последнем. Просто чем больше тираж продукта, тем лучше он отлажен.

10. После разработки плана по изоляции ошибки следует ранжировать (ранг; сортировка) гипотезы по вероятности их подтверждения. Начинать проверку целесообразно не с самой вероятной гипотезы, а с той, которую можно быстрее всего проверить. Тем самым можно быстро отсечь часть гипотез и сузить процесс проверки.

11. Затем гипотезы проверяются по очереди (строго по одной в единицу времени), в определенной последовательности. В восходящем направлении – от рабочей станции к коммутационной аппаратуре или серверу либо в нисходящем направлении – от сервера или коммутационной аппаратуры к рабочей станции. Для проверки используются только специальные проверенные версии программных продуктов, специальные тестовые кабели и проверенные надежные тестовые диагностические средства.

12. Наконец, последним действием является документирование проблемы и способа ее решения в специальном журнале. Обязательно должны быть созданы инструкции службам АС по действиям, предотвращающим повторное появление проблемы.

**20. Стратегии определения ошибок при администрировании ИС. Два подхода к поиску неисправностей. Типы стратегий. Технологии работы NMS.**

Существуют **два подхода** к поиску неисправностей – теоретический и практический.

При теоретическом подходе специалист-теоретик анализирует ситуацию до тех пор, пока не будет найдена точная причина ошибки.

• При таком решении, например, сетевой проблемы требуется современный высокопроизводительный протокольный анализатор для набора и анализа огромного количества сетевого трафика в течение значительного времени.

• Затем сетевому специалисту необходим длительный теоретический анализ данных.

• Этот процесс надежен, однако не многие компании могут себе позволить, чтобы их ИС или сеть не функционировала в течение нескольких часов или даже дней.

При практическом подходе опыт специалиста-практика подсказывает, что при возникновении неисправности целесообразно начинать менять сетевые платы, кабели, аппаратные средства и программное обеспечение до тех пор, пока система не начнет работать.

• Это вовсе не означает, что все компоненты системы функционируют должным образом, главное, что они вообще функционируют.

• К сожалению, во многих руководствах по эксплуатации в разделе поиска неисправностей фактически рекомендуется прибегнуть к стилю специалиста-практика, вместо предоставления подробной инструкции по устранению технических неисправностей.

• Этот подход быстрее предыдущего. Однако он очень ненадежен и первопричина неработоспособности системы может быть так и не устранена

Ни тот, ни другой метод чаще всего не дают желаемых результатов при поиске и устранении неисправностей. Поэтому действия администратора системы должны базироваться на стратегии управления ошибками. **Стратегия управления** ошибками может быть проактивной либо реактивной.

• С ростом объема ИС возрастает потребность в ее надежности и, соответственно, возрастает потребность в предварительном мониторинге производительности системы, предупреждениях пользователям о возможных проблемах, постоянной бдительности администратора системы. Такая стратегия предупреждения ошибок называется проактивной.

• Стратегия, при которой АС не предупреждает появление ошибок, а разбирается с ошибками по мере их возникновения, называется реактивной.

АС должен приложить усилия и воспользоваться средствами MS (специализированное ПО, которое ведет журнал ошибок, собирает статистику, фиксирует конфигурации средств системы, опознает тревожные ситуации, только помогает АС, а не устраняет аппаратные или кабельные проблемы) или NMS (для управления только сетевыми системами (Network Management System), реализации своих функций использует протокол SNMP) для перехода от реактивной стратегии к проактивной.

Обычно системы управления отказами (ошибками) – NMS разбивают сложную задачу идентификации и диагностики ошибки на четыре подзадачи:

• определение ошибки;

• генерация тревожного сигнала;

• изоляция ошибки;

• коррекция ошибки.

При этом возможны **две технологии** работы NMS – пассивная и активная.

Пассивная технология. С помощью протокола SNMP устройства оповещают управляющую систему о выполнении заранее предусмотренного и заданного параметрами системы условия, например, отличие какого-либо параметра от номинального значения. Эта технология должна применяться АС при идентификации проблем, не связанных с аппаратными сбоями, например, при изменении производительности, проблемах интерфейсов и т. д.

Активная технология. Система NMS тестирует ИС (например, с помощью утилиты PING) и опрашивает каждое из устройств на регулярной основе. Если какое-либо устройство не реагирует в заданный АС интервал времени или его параметры отличаются от желаемых, посылается сообщение АС о сбое устройства.

АС должен выбрать систему управления, позволяющую использовать обе стратегии, правильно спроектированная система управления дает возможность АС выполнять далее перечисленные логические действия по управлению ошибками.

**3 блок вопросов**

**1. Системы HelpDesk. Системы ServiceDesk.**

В ИТ инфраструктуре для максимально быстрой ликвидации проблем – аварий, проблем с оборудование, и тд. Создаются специальные отделы, которые контактируют с сотрудниками и согласовывают ликвидацию проблем с ИТ подразделеним. Этот отдел может называться Центр обслуживания пользователей (Service Desk) или Центр поддержки пользователей (Help Desk).

Задачей Service Desk является регистрация заявок пользователей, предоставление им требуемой помощи и привлечение сотрудников ИТ подразделения для скорейшего устранения проблем. Дополнительно эта служба анализирует статистику инцидентов и время их устранения. Это необходимо для оценки и повышения качества предоставления ИТ услуг.

ServiceDesk, помимо решения инцидентов, также управляет запросами на обслуживание (о предоставлении новых услуг) и запросами о предоставлении информации (отвечает на вопросы «как мне это сделать?»).

Help desk – более узкое понятие, это инструмент технической поддержки пользователей.

Helpdesk появилась как ИТ-ориентированная служба для решения внутренних проблем, а ServiceDesk – услугоориентированная служба, которая развивает концепцию предоставления ИТ как услуг бизнесу. Причем услуги не только ИТ-услуги, ServiceDesk может выполнять функцию общего центра обслуживания для приема и обработки запросов в хозяйственную, финансовую или юридическую службу организации.

**2. Протоколы POP3, IMAP, SMTP. Методы шифрования SSL, TLS. Форматы почтовых сообщений.**

Post Office Protocol Version 3 - протокол почтового отделения, версия 3 — это сетевой протокол, используемый почтовым клиентом для получения сообщений электронной почты с сервера. Обычно используется в паре с протоколом SMTP. По умолчанию использует TCP-порт 110.

После установки соединения протокол РОР3 проходит три последовательных состояния

* Авторизация клиент проходит процедуру аутентификации
* Транзакция клиент получает информацию о состоянии почтового ящика, принимает и удаляет почту.
* Обновление сервер удаляет выбранные письма и закрывает соединение.

IMAP (англ. Internet Message Access Protocol) – интернет-протокол прикладного уровня для доступа к электронной почте.

IMAP предоставляет пользователю богатые возможности для работы с почтовыми ящиками, находящимися на центральном сервере. Почтовая программа, использующая этот протокол, получает доступ к хранилищу корреспонденции на сервере так, как будто эта корреспонденция расположена на компьютере получателя. Электронными письмами можно манипулировать с компьютера пользователя (клиента) без необходимости постоянной пересылки с сервера и обратно файлов с полным содержанием писем.

SMTP (Simple Mail Transfer Protocol) – это сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP, причем передача должна быть обязательно инициирована самой передающей системой.

TLS ( англ. Transport Layer Security) — криптографический протокол, обеспечивающий безопасную передачу данных между пользователями в сети Интернет. TLS-протокол основан на Netscape SSL-протоколе версии 3.0 и состоит из двух частей — TLS Record Protocol и TLS Handshake Protocol. Различие между SSL 3.0 и TLS 1.0 незначительные.

SSL (англ. Secure Sockets Layer — протокол защищённых сокетов) — криптографический протокол, обеспечивающий безопасную передачу данных по сети Интернет. При его использовании создаётся защищённое соединение между клиентом и сервером.

**3. Система доменных имен DNS. Протоколы TELNET, FTP.**

DNS по английский Domain Name System — это протокол прикладного уровня стека протоколов tcp-ip.

Система доменных имен (DNS, Domain Name System) представляет собой механизм, а именно распределенную БД, предназначенный для поиска по имени хоста его IP адреса и некоторой другой информации.

Смысл её в том, что каждому цифровому IP-адресу присваивается понятное буквенное имя (домен). Когда вы вводите в браузере доменное имя, сервера DNS преобразуют его в IP-адрес.

TELNET (сокр. от англ. teletype network) — сетевой протокол для реализации текстового терминального интерфейса по сети (в современной форме — при помощи транспорта TCP).

Выполняет функции протокола прикладного уровня модели OSI. Протокол telnet, наряду с ssh широко используется для удалённого администрирования различными сетевыми устройствами и программными серверами.

FTP (File Transfer Protocol - протокол передачи файлов) – это популярный способ передачи файлов между двумя удаленными системами, например между Вашим компьютером и Вашим хостингом

**4. Семиуровневая модель сетевого обмена. Стек протоколов TCP.**

Модель OSI:

1. Физический уровень: работа со средой передачи, сигналами и двоичными данными. Примеры: USB, витая пара, коаксиальный кабель, оптоволоконный кабель, радиоканал
2. Канальный уровень: физическая адресация. MAC-адрес. На этом уровне обитают такие устройства как коммутаторы и мосты.
3. Сетевой уровень: определение маршрута и логическая адресация. IP – адреса.
4. Транспортный уровень: обеспечивает передачу данных по сети интернет. TCP и UDP. TCP (Transmission Control Protocol)! Он обеспечивает контроль за передачей данных, протокол без потерь. UDP (User Datagram Protocol), протокол с потерями.
5. Сеансовый уровень: управлением сеансом связи. Управляет соединениями, или попросту говоря, сессиями. Он их разрывает.
6. Представительский уровень: шифрование данных. Представление (кодировка данных). Преобразование форматов сообщений, такое как кодирование или сжатие. Тут живут JPEG и GIF, например. Так же уровень ответственен за передачу потока на четвертый (транспортный уровень).
7. Прикладной уровень: доступ к сетевым службам. Тут находятся сетевые службы, которые позволяют нам, как конечным пользователям, серфить просторы интернета.

Стек протоколов TCP/IP – это организованный набор основных сетевых протоколов, который иерархическим способом разделен на четыре уровня и представляет собой систему транспортного распределения пакетов по компьютерной сети.

Любой протокол передачи данных TCP/IP для идентификации узлов использует один из следующих типов адресов: Локальные (аппаратные) адреса. Сетевые адреса (IP адреса). Доменные имена.

**5. Реестр Windows. Службы ОС в Windows.**

Реестр – это иерархически построенная база данных параметров и настроек в большинстве операционных систем Microsoft Windows. Реестр содержит информацию и настройки для аппаратного обеспечения, программного обеспечения, профилей пользователей, предустановки. Большинство изменений в Панели управления, ассоциации файлов, системные политики, список установленного ПО фиксируются в реестре.

Файлы реестра хранятся на системном диске в папке Windows/System32/Config — файлы SAM, SECURITY, SYTEM и SOFTWARE содержат информацию из соответствующих разделов в HKEY\_LOCAL\_MACHINE. Данные из HKEY\_CURRENT\_USER хранятся в скрытом файле NTUSER.DAT в папке «Users/Имя\_пользователя» на компьютере.

**HKEY\_CURRENT\_USER**: Данный раздел является корневым для данных конфигурации пользователя, вошедшего в систему в настоящий момент. Здесь хранятся папки пользователя, цвета экрана и параметры панели управления. Эти сведения сопоставлены с профилем пользователя. Вместо полного имени раздела иногда используется аббревиатура HKCU.

**HKEY\_USERS:** Данный раздел содержит все активные загруженные профили пользователей компьютера. Раздел HKEY\_CURRENT\_USER является подразделом раздела HKEY\_USERS. Вместо полного имени раздела иногда используется аббревиатура HKU.

**HKEY\_LOCAL\_MACHINE:** Раздел содержит параметры конфигурации, относящиеся к данному компьютеру (для всех пользователей). Вместо полного имени раздела иногда используется аббревиатура HKLM.

**HKEY\_CURRENT\_CONFIG:** Данный раздел содержит сведения о профиле оборудования, используемом локальным компьютером при запуске системы.

**HKEY\_CLASSES\_ROOT:** Является подразделом HKEY\_LOCAL\_MACHINE\Software. Хранящиеся здесь сведения обеспечивают выполнение необходимой программы при открытии файла с использованием проводника. Вместо полного имени раздела иногда используется аббревиатура HKCR.

Службы Windows – специальные программы, запускаемые в фоновом режиме для обеспечения согласованной работы аппаратных и софтовых компонентов компьютера, либо для поддержки тех или иных системных функций.

В различных версиях операционных систем могут присутствовать одни службы и отсутствовать другие. Некоторые приложения и программы, устанавливаемые отдельно, также могут создавать свои службы.

Windows предлагает диспетчер управления службами (Service Control Manager — SCM), с помощью которого можно управлять созданием, удалением, запуском и остановкой служб. Приложение, имеющее статус службы, должно быть написано таким образом, чтобы оно могло принимать сообщения от SCM. Затем имя службы и её другие атрибуты, например, описание службы, регистрируются в диспетчере управления при помощи одного или нескольких вызовов API.

Список основных служб операционных систем Microsoft:

* Dhcp

Регистрирует и обновляет IP-адреса и DNS-записи для этого компьютера. Если эта служба остановлена, этот компьютер не сможет получать динамические IP-адреса и выполнять обновления DNS.

* Dnscache

Служба DNS-клиента (dnscache) кэширует имена DNS (Domain Name System) и регистрирует полное имя данного компьютера. Если служба остановлена, разрешение имен DNS будет продолжаться. Однако результаты очередей имен DNS не будут кэшироваться, и имя компьютера не будет зарегистрировано.

* KtmRm

Координирует транзакции между MSDTC и диспетчером транзакций ядра (Kernel Transaction Manager - KTM).

* EMDMgmt

Поддержка повышения производительности системы с помощью технологии ReadyBoost.

* SysMain

Поддерживает и улучшает производительность системы.

* Audiosrv

Управление средствами работы со звуком для программ Windows. Если эта служба остановлена, то аудиоустройства и эффекты не будут правильно работать.

* Idsvc

Обеспечивает надежную возможность создания, управления и раскрытия цифровых удостоверений.

* WUAUSERV

Включает загрузку и установку обновлений Windows. Если служба отключена, то на этом компьютере нельзя будет использовать возможности автоматического обновления или веб-узел Windows Update.

* RpcSs

Обеспечивает сопоставление конечных точек и иных служб RPC.

Для каждой службы можно поменять тип ее запуска. Сделать это можно, кликнув правой кнопкой мыши по службе и нажав на «Открыть службы». Там снова нажимаем правую кнопку мыши, но в этот раз на нужной нам службе и выбираем «Свойства».

**6. Язык сценариев Power Shell. Командлеты в PowerShell. Примеры. Конвейер в PowerShell. Примеры.**

PowerShell — это кроссплатформенная система для автоматизации задач и управления конфигурацией, состоящая из оболочки командной строки и языка сценариев. В отличие от большинства оболочек, которые принимают и возвращают текст, PowerShell построена на основе общеязыковой среды выполнения (CLR) .NET и принимает и возвращает объекты .NET. Это фундаментальное отличие, которое подразумевает использование совершенно новых средств и методов автоматизации.

С помощью Windows PowerShell можно:

* Получать доступ к файловой системе;
* Управлять реестром;
* Управлять службами;
* Управлять процессами;
* Настраивать операционную систему;
* Устанавливать программное обеспечение;
* Устанавливать роли и компоненты сервера;
* Осуществлять администрирование и конфигурирование ролей и компонентов сервера;
* Писать и использовать сценарии для автоматизации управления и администрирования;
* Выполнять другие задачи системных администраторов.

Командлет (cmdlet) – это команда Windows PowerShell, с помощью которой можно осуществлять взаимодействие с объектами операционной системы с целью их управления. Данные команды являются частью языка PowerShell. Командлеты построены по принципу «Глагол-Существительное», разделенные дефисом (-).

Полный список командлетов в Windows PowerShell можно посмотреть с помощью специального командлета Get-Command.

Пример командлета   
Get-CimInstance – командлет, который позволяет обращаться к различным классам.  
Select-Object – фильтрует вывод информации.  
ExcludeProperty – помогает вывести нужные колонки.  
Start-Process – этот командлет позволяет запустить указанный файл в виде процесса.   
С помощью -FilePath мы указываем элемент, который нам нужно запустить

Конвейер – это передача результата работы командлета через вертикальную черту (|) другому командлету.

С помощью конвейера можно выполнять сложные задачи простым и удобным способом без необходимости написания сложных алгоритмов и сценариев.

Пример

Get-CimInstance -ClassName Win32\_Processor | Select-Object -ExcludeProperty "CIM\*"

**7. Системы удаленного доступа к компьютеру и управлению программным обеспечением.**

**Системы удаленного доступа -** [Программы](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80)) или функции [операционных систем](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), позволяющие получить удалённый доступ к компьютеру через [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82) или [ЛВС](https://ru.wikipedia.org/wiki/%D0%9B%D0%92%D0%A1) и производить управление и [администрирование](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%BE%D0%B5_%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) удалённого компьютера [в реальном времени](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B6%D0%B8%D0%BC_%D1%80%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%B8). Программы удалённого администрирования предоставляют почти полный [контроль](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C) над удалённым компьютером: они дают возможность удалённо управлять рабочим столом и всей операционной системой компьютера, возможность копирования или удаления файлов, установки и запуска приложений и т. д.

LiteManager, team viewer.

**8. Системы конфигурирования ИС. Сиcтема управления версиями GIT (цели использования git, принцип работы, репозиторий, типы репозиториев).**

В ходе разработки ПО требуется постоянно вести учёт изменений исходного кода, а также возможность отката к контрольной точке. Для этого были созданы специальные инструменты — системы управления версиями, к коим и относится рассматриваемое ПО git.

Основное отличие Git от любой другой СКВ (включая Subversion и её собратьев) — это подход к работе со своими данными. Концептуально, большинство других систем хранят информацию в виде списка изменений в файлах. Эти системы (CVS, Subversion, Perforce, Bazaar и т.д.) представляют хранимую информацию в виде набора файлов и изменений, сделанных в каждом файле, по времени (обычно это называют контролем версий,

Git не хранит и не обрабатывает данные таким способом. Вместо этого, подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы. Каждый раз, когда вы делаете коммит, то есть сохраняете состояние своего проекта в Git, система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок. Для увеличения эффективности, если файлы не были изменены, Git не запоминает эти файлы вновь, а только создаёт ссылку на предыдущую версию идентичного файла, который уже сохранён. Git представляет свои данные как **поток снимков**.

**Репозиторий Git** представляет собой каталог файловой системыю

В классическом обычном сценарии в репозитории git есть три типа объектов — файл, дерево и «коммит» ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *commit* — фиксация). Файл есть какая-то версия какого-то пользовательского файла, дерево — совокупность файлов из разных подкаталогов, «коммит» — дерево и некая дополнительная информация (например, родительские коммиты, а также комментарий).

Репозиторий Git бывает локальный и удалённый. Локальный репозиторий — это подкаталог .git, создаётся (в пустом виде) командой git init и (в непустом виде с немедленным копированием содержимого родительского удалённого репозитория и простановкой ссылки на родителя) командой git clone.

**9. Управление пакетами (dpkg, apt-get, Aptitude).**

**Dpkg** – это пакетный менеджер для Debian систем. Он может устанавливать, удалять и создавать пакеты, но, в отличие от других систем управления пакетами, он не может автоматически загружать и устанавливать пакеты или их зависимости. Эта секция раскрывает использование dpkg для управления локально установленными пакетами.

**Apt-get** – это мощный консольный инструмент, который работает с Улучшенным инструментарием пакетов (APT) Ubuntu, выполняющий такие функции, как установка новых программных пакетов, обновление имеющихся пакетов, обновления индекса списка пакетов и даже обновление все системы Ubuntu.

Будучи простым консольным инструментом, apt-get имеет ряд преимуществ перед другими инструментами управления пакетами, доступными в Ubuntu серверным администраторам. Некоторые из этих преимуществ включают легкое использование простых терминальных соединений (SSH) а также возможность использования в сценариях системных администраторов, которые могут быть автоматизированы с помощью утилиты планирования cron.

**Aptitude** с данным менеджером можно взаимодействовать как через командную строку, аналогично apt-get, так и с помощью псевдографического интерфейса, с помощью которого управлять пакетами намного удобнее.

**10. Интерфейсы Ethernet (выполнить определение Ethernet интерфейсов, логическое имя, настройка интерфейса).**

Интерфейсы Ethernet идентифицируются системой с использованием имен ethX, где X является числовым значением. Первый интерфейс обычно обозначается как eth0, второй как eth1, и все последующие с увеличивающимися номерами по порядку.

**11. Адресация IP (временное назначение IP адреса, динамическое присвоение и статическое присвоение IP адреса, интерфейс Loopback – обратная петля).**

Схема IP-адресации, применяемая в **TCP/IP**, позволяет пользователям и приложениям однозначно идентифицировать сети и хосты, с которыми устанавливаются соединения.

IP-адрес работает так же, как и почтовый адрес, позволяя направлять данные в выбранный пункт назначения. Протокол **TCP/IP** описывает стандарты для присвоения адресов сетям, подсетям, хостам, сокетам, а также для применения специальных адресов оповещения и локальных циклических адресов.

IP-адрес состоит из адреса сети и адреса хоста (или локального адреса). Такой адрес, состоящий из двух частей, позволяет отправителю задавать как сеть, так и конкретный хост в этой сети. Каждой сети присваивается уникальный адрес при подсоединении ее к другим сетям Internet. Однако, если вы не планируете подключать локальную сеть к другим сетям Internet, ей можно присвоить любой сетевой адрес.

Временное назначение IP адреса.

Для временной настройки сети вы можете использовать стандартные команды, такие как ip, ifconfig и route, которые присутствуют также и в других системах на базе GNU/Linux. Эти команды позволят изменить настройки, которые будут применены мгновенно, но они не будут постоянными и будут утеряны после перезагрузки.

Команды для временного назначения IP адреса:

1. Ifconfig – для проверки настройки IP адреса eth0
2. Route – для настройки шлюза по умолчанию
3. ip с опцией flush – для того, чтобы отменить все IP настройки интерфейса

**Loopback** — это термин, который обычно используется для описания методов или [процедур](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) [маршрутизации](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) электронных сигналов, цифровых [потоков данных](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), или других движущихся сущностей от их источника и обратно к тому же источнику без специальной обработки или модификаций. Первоначально он использовался для тестирования передачи или передающей инфраструктуры.

Примеры использования:

* *метод проведения теста передачи по линии доступа от обслуживающего центра коммутации*, для которого обычно не требуется поддержка персонала на обслуживаемом терминале;
* *метод тестирования передачи сигнала между станциями* (необязательно соседними), где используется две линии передачи, тестирование проходит на одной станции и две линии связывают её со станцией на расстоянии. Обычно называется loop around, когда для доступа к связывающей цепи нужно набрать номера или код;
* *коммутационный кабель* применяется вручную или автоматически, удаленно или локально, что облегчает loop-back тест;
* *коммуникационный канал с единственной конечной точкой*. Любые сообщения передаваемые через такой канал немедленно получаются тем же каналом;
* *возвратная петля, обратная петля* — вид диагностического теста, при котором сигнал возвращается передающему устройству, пройдя по коммуникационному каналу в обоих направлениях. (адрес возвратной петли используется в качестве идентификатора маршрутизаторов Cisco для работы с протоколом OSPF. [ RouterX(config)# interface loopback <*number*> ].

**12. TCP/IP (IP адрес, сетевая маска, адрес сети, адрес рассылки, адрес шлюза, адрес сервера имен). TCP. UDP.ICMP**

Протокол управления передачей и межсетевой протокол (**TCP/IP**) — это стандартный набор протоколов, разработанный в конце 1970-х Агентством передовых оборонных исследовательских проектов (DARPA) как средство взаимодействия между различными типами компьютеров и компьютерными сетями. TCP/IP является движущей силой интернета и поэтому является самым популярным стеком сетевых протоколов на Земле.

**ICMP**

Протокол межсетевых управляющих сообщений (ICMP) — это расширение протокола IP, определенное в RFC 792, поддерживающее сетевые пакеты сообщений управления, информирующих сообщений и сообщений об ошибках. ICMP используется в таких приложениях как утилита ping, которая определяет доступность сетевого хоста или устройства. Примерами некоторых сообщений об ошибках, возвращаемых ICMP, которые полезны как для сетевых хостов, так и роутеров, являются «адресат недоступен» (Destination Unreachable) и «время вышло» (Time Exceeded).

Пользовательский пакетный протокол (**UDP**) — это протокол без установления соединения, который редко используется для передачи важных данных, поскольку не использует управление потоком и иные методы проверки целостности передаваемых данных. UDP обычно используется в таких приложениях, как передача аудио- и видео- потоков, поскольку он значительно быстрее TCP из-за отсутствия коррекции ошибок и управления потоком, где потеря нескольких пакетов обычно не катастрофично.

* 1. **IP адрес** – это уникальная идентифицирующая строка, представленная в виде четырех десятичных чисел от 0 до 255, разделенных точками, каждое из которых представляет 8-битный адрес, составляющие полный 32-битный адрес Этот формат называют точечной четверичной записью (dotted quad notation).
  2. **Маска подсети (или просто сетевая маска)** — это локальная побитовая маска или набор флагов, которые разделяют IP адрес на часть, значимую для всей сети, и биты, значимые для данной подсети. Например, в сети класса C стандартная сетевая маска 255.255.255.0, которая маскирует первые 3 байта IP адреса и делает последний байт адреса доступным для хостов, определенных в подсети.
  3. **Адрес сети** представляет собой байты, составляющие сетевую часть IP адреса. Например, хост 12.128.1.2 в сети класса A будет использовать 12.0.0.0 в качестве адреса сети, где 12 представляет первый байт IP адреса (сетевая часть) и нули в оставшихся трех байтах для предоставления возможных значений для хостов. Сетевой хост, использующий частный IP адрес 192.168.1.100 будет использовать для адреса сети 192.168.1.0, который определяет первые три байта сети 192.168.1 класса C и 0 для всех возможных хостов сети.
  4. **Адрес рассылки** — это IP адрес, который позволяет посылать сетевые данные на все хосты заданной подсети вместо того чтобы задавать определенный хост. Стандартный общий адрес рассылки для IP сетей 255.255.255.255, но этот адрес рассылки не может быть использован для отправки сообщения каждому хосту в интернете, поскольку роутеры его блокируют. Более уместно устанавливать адрес рассылки, соответствующий вашей подсети. Например, в частной IP сети класса С 192.168.1.0 адрес рассылки будет 192.168.1.255. Широковещательные сообщения обычно создаются сетевыми протоколами, такими как протокол определения адреса (ARP) и информационный протокол маршрутизации (RIP).
  5. **Адрес шлюза** — это IP адрес, через который можно соединиться с определенной сетью или хостом в сети. Если один сетевой хост хочет соединиться с другим хостом в сети, и этот хост находится не в этой же сети, необходимо использовать шлюз. В большинстве случаев адрес шлюза является адресом роутера в этой сети, который включается для пропускания трафика к другим сетям или хостам, таким как интернет хостам. Значение адреса шлюза должно быть корректным или система не сможет получать доступ к хостам за пределами этой самой сети.
  6. **Адрес сервера имен** представляет IP адрес системы сервиса доменных имен (DNS), которые разрешают сетевые имена хостов в IP адреса. Существует три уровня адресов серверов имен, которые могут быть определены в порядке старшинства: Первичный сервер имен, Вторичный сервер имен и Третичный сервер имен. Чтобы у вашей системы была возможность разрешения сетевых имен хостов в соответствующие им IP адреса, вы должны определить правильные адреса серверов имен, которые вам разрешено использовать в настройках TCP/IP вашей системы. В большинстве случаев эти адреса могут и предоставляются вашим сетевым провайдером, но есть много свободных и публично доступных серверов имен, которые можно использовать, таких как Level3 (Verizon) сервера с адресами от 4.2.2.1 до 4.2.2.6.

**13. Протокол динамического выделения адресов (DHCP). NTP протокол.**

Протокол динамического выделения адресов (DHCP) — это сетевой сервис, который позволяет компьютерам в сети автоматически получать настройки с сервера вместо того, чтобы настраивать каждый сетевой хост вручную. Компьютеры, настроенные быть клиентами DHCP, не управляют тем, какие настройки они получат от DHCP сервера, и эта настройка совершенно незаметна для пользователя компьютера.

В общем случае настройки, передаваемые DHCP сервером DHCP клиентам, включают: IP адрес и сетевую маску, IP адрес шлюза по умолчанию, IP адрес DNS серверов.

Однако DHCP сервер может также предоставить такие параметры настройки, как: имя хоста, имя домена, адрес сервера времени, адрес сервера печати.

Преимущество использования DHCP заключается в изменчивости сети, например, изменение адреса DNS сервера потребует изменений только на DHCP сервере, а все сетевые хосты будут перенастроены в момент следующего запроса их DHCP клиента к DHCP серверу. Дополнительное преимущество заключается в простом подключении новых компьютеров к сети, поскольку не требуется проверять доступность IP адресов. Конфликты по выделенным IP адресам также минимальны.

DHCP сервер может предоставлять настройки, используя следующие методы:

**Выделение вручную (по MAC адресу)**

Этот метод подразумевает использование DHCP для определения уникального аппаратного адреса каждой сетевой карты, подключенной к сети, и затем продолжительного предоставления неизменной конфигурации каждый раз, когда DHCP клиент делает запрос на DHCP сервер, используя это сетевое устройство. Это гарантирует, что определенный адрес будет автоматически присваиваться этой сетевой карте на основе ее MAC адреса.

**Динамическое выделение (пул адресов)**

При этом методе DHCP сервер будет выделять IP адрес из пула адресов (иногда называемым диапазоном или областью) на период времени (или в аренду), который настраивается на сервере, или пока клиент не проинформирует сервер, что больше вообще не нуждается в адресе. Таким образом, клиенты получают свои настройки динамически по принципу «первый пришел - первый обслужился». Когда DHCP клиент отсутствует в сети определенное время, настройка считается просроченной и возвращается в пул адресов для использования другими DHCP клиентами. Это означает, что адрес арендуется или выдается на определенный период времени. По истечении этого периода клиент должен повторно договариваться об использовании адреса с сервером.

**Автоматическое выделение**

Использую этот метод, DHCP автоматически присваивает постоянный IP адрес устройству, выбранный из пула доступных адресов. Обычно DHCP используется для выдачи временного адреса, но DHCP сервер может использовать бесконечное время аренды.

**NTP** — это протокол синхронизации времени по сети. По существу клиенты запрашивают текущее время на сервере и используют его для установки своих собственных часов.

За этим простым описанием скрывается много сложностей - существуют уровни NTP серверов, где первый уровень подключен к атомным часам, а второй и третий уровни серверов распределяют на себя нагрузку по актуальным запросам из интернета. Кроме того, клиентское приложение сложнее, чем вы можете подумать - оно компенсирует задержки соединения и регулирует время таким образом, чтобы не навредить другим процессам, запущенным на сервере.

**14. Администрирование баз данных в UbuntuServer (MySQL, PostgreSQL).**

Одной из самых популярных СУБД на сегодняшний день является **MySQL,** распространяемая свободно (с некоторыми ограничениями). Эта серверная система способна эффективно функционировать во взаимодействии с интернет-сайтами и веб-приложениями. При этом она проста в освоении, что лишь увеличивает ее популярность.

Помимо универсальности и распространенности СУБД MySQL обладает целым комплексом важных преимуществ перед другими системами. В частности следует отметить такие качества как:

* Простота в использовании. MySQL достаточно легко инсталлируется, а наличие множества плагинов и вспомогательных приложений упрощает работу с базами данных.
* Обширный функционал. Система MySQL обладает практически всем необходимым инструментарием, который может понадобиться в реализации практически любого проекта.
* Безопасность. Система изначально создана таким образом, что множество встроенных функций безопасности в ней работают по умолчанию.
* Масштабируемость. Являясь весьма универсальной СУБД, MySQL в равной степени легко может быть использована для работы и с малыми, и с большими объемами данных.
* Скорость. Высокая производительность системы обеспечивается за счет упрощения некоторых используемых в ней стандартов.

Как и любой программный продукт, система MySQL имеет определенные ограничения в своем функционале, что не позволяет использовать ее для работы с приложениями, имеющими некоторые специфические требования. К недостаткам этой СУБД относятся:

* Недостаточная надежность. В вопросах надежности некоторых процессов по работе с данными (например, связь, транзакции, аудит) MySQL уступает некоторым другим СУБД.
* Низкая скорость разработки. Как и многим другим программным продуктам с открытым кодом, MySQL не достает некоторого технического совершенства, что порой сказывается на эффективности процессов разработки.

**PostgreSQL** – унифицированный сервер баз данных, имеющий единый движок – storage engine. Постгрес использует клиент-серверную модель.

Для каждого клиента на сервере создается новый процесс (не поток). Для работы с такими клиентскими процессами сервер использует семафоры.

Клиентский запрос проходит следующие стадии.

1. Коннект.
2. Парсинг: проверяется корректность запроса и создается дерево запроса (query tree). В основу парсера положены базовые юниксовые утилиты yacc и lex.
3. Rewrite: берется дерево запросов и проверяется наличие в нем правил (rules), которые лежат в системных каталогах. Всякий раз пользовательский запрос переписывается на запрос, получающий доступ к таблицам базы данных.
4. Оптимизатор: на каждый запрос создается план запроса – query plan, который передается исполнителю – executor. Смысл плана в том, что в нем перебираются все возможные варианты получения результата (использовать ли индексы, джойны и т.д.), и выбирается самый быстрый вариант.
5. Выполнение запроса: исполнитель рекурсивно проходит по дереву и получает результат, используя при этом сортировку, джойны и т.д., и возвращает строки. Постгрес – обьектно-реляционная база данных, каждая таблица в ней представляет класс, между таблицами реализовано наследование. Реализованы стандарты SQL92 и SQL99.
6. Транзакционная модель построена на основе так называемого multi-version concurrency control (MVCC), что дает максимальную производительность. Ссылочная целостность обеспечена наличием первичных и вторичных ключей.

Как и любая СУБД PostgreSQL должна обслуживать запросы sql и возвращать результаты клиентским приложениям. Для этого механизм СУБД оптимизатор запросов принимает решение об оптимальном использовании ресурсов и строит план запроса. При этом он может опереться на используемых механизмы ускорения работы (индексов, кэшей в памяти, ресурсам процессора, данные статистики и т.д.). Результаты запросов возвращаются клиентским подключениям.

Сильными сторонами PostgreSQL считаются:

* поддержка БД практически неограниченного размера;
* мощные и надёжные механизмы транзакций и репликации;
* расширяемая система встроенных языков программирования: в стандартной поставке поддерживаются PL/pgSQL, PL/Perl, PL/Python и PL/Tcl; дополнительно можно использовать PL/Java, PL/PHP, PL/Py, PL/R, PL/Ruby, PL/Scheme и PL/sh, а также имеется поддержка загрузки C-совместимых модулей;
* наследование;
* легкая расширяемость.

**15. Средства безопасности в Ubuntu Server. Управление пользователем. Безопасность консоли. Firewall. AppArmor**

**Firewall -** главным образом обеспечение безопасности локальных сетей зависит от программных средств. К таковым относятся:

* Межсетевые экраны. Это промежуточные элементы компьютерной сети, которые служат для фильтрации входящего и исходящего трафика. Риск несанкционированного доступа к информации становится меньше.
* Прокси-серверы. Производят ограничение маршрутизации между глобальной и локальной частями сети.
* VPN. Позволяют передавать информацию по зашифрованным каналам.
* Разные наборы протоколов, которые нужны для создания защищенного соединения и установления контроля над элементами локальной сети.

Эти приложения, встроенные в оперативную систему и специализированные, шифруют данные. Данные разграничивают потоки информации.

К средствам физической защиты относят систему архивирования и размножения информации. Для крупномасштабных корпоративных сетей рекомендовано организовывать отдельный архивационный сервер.

**AppArmor** – это реализация Модуля безопасности Linux по управлению доступом на основе имен. Он устанавливается и загружается по умолчанию. Принцип работы Apparmor: он использует профили приложений для определения какие файлы и права доступа требуются приложению, поэтому необязательно полностью включать или отключать Apparmor, можно контролировать доступ только для отдельных приложений. Некоторые пакеты устанавливают свои собственные профили, а дополнительные профили можно найти в пакете apparmor-profiles.

Ядро линукс включает подсистему Netfilte, который используется для манипулирования или решения судьбы сетевого трафика, передаваемого в или через ваш сервер. Все современные решения линукс по сетевой защите используют эту систему пакетной фильтрации.

Система пакетной фильтрации на уровне ядра была бы малоиспользуема администраторами без пользовательского интерфейса для ее управления. Для этого предназначен iptables. Когда пакет попадает на ваш сервер, он передается подсистеме Netfilter для одобрения, изменения или отказа на основе правил, которые она получает от интерфейса пользователя через iptables. Таким образом iptables - это все, что вам нужно для управления вашей сетевой защитой, если вы хорошо с ним знакомы, однако множество внешних интерфейсов доступны для упрощения этой задачи.

Инструмент для настройки сетевой защиты Ubuntu по умолчанию – это ufw. Он разработан для легкой настройки iptables и предоставляет дружественный способ создания сетевой защиты для IPv4 и IPv6.