**Глава 27. Служба удаленного доступа**

Служба удаленного доступа предоставляет авторизованным лицам возможность подключаться к сети компании из дома, удаленных местоположений или даже с другого континента. Ранее эта услуга была нужна только техническим специалистам для работы вне обычного рабочего времени, но сегодня она стала базовой услугой, которой пользуются все сотрудники компании, включая удаленных работников и командировочных.

Удаленный доступ может быть организован различными способами, и можно выделить две основные категории. В первой категории подключение осуществляется напрямую к сети через телефонные модемы, ISDN и другие аналогичные технологии. Во второй категории используется подключение через Интернет (WiFi, кабельные модемы, DSL, Ethernet и другие), после чего доступ осуществляется через туннель или VPN.

Удаленный доступ связан с рядом проблем. Во-первых, пользователи хотят иметь возможность проверять электронную почту и работать с данными, когда они находятся в пути. Во-вторых, существует потребность у некоторых работников в том, чтобы работать из дома по вечерам, в выходные или в другие дни недели. Еще один важный аспект — это необходимость постоянного доступа к корпоративной сети для тех, кто часто находится в удаленных локациях. Эти три потребности имеют общие черты, но требуют разных решений.

Сфера удаленного доступа находится в постоянном изменении, и технологии продолжают развиваться. В этой главе рассматриваются основные принципы создания и администрирования службы удаленного доступа, без углубления в подробности технологий, которые могут изменяться в процессе времени. Информация, представленная в главе, поможет вам оценить текущие технологии и подходы, а также сделать осознанные архитектурные решения.

**27.1. Основы**

Для создания службы удаленного доступа важно понимать разнообразие требований ваших пользователей. Также необходимо совместно с пользователями определиться с уровнями обслуживания, которые будут предоставляться системными администраторами для различных аспектов системы, и документировать эти решения для использования в будущем.

Когда требования и уровни обслуживания определены, можно принимать решение о создании службы или об отказе от нее. Одним из главных принципов создания службы удаленного доступа является возможность передачи части задач сторонним исполнителям. Однако есть компоненты, которые должны быть организованы и контролироваться внутри компании, особенно это касается аспектов безопасности, таких как аутентификация, авторизация и поддержание безопасности периметра сети.

**27.1.1. Требования к удаленному доступу**

Первое требование к службе удаленного доступа, скорее всего, будет восприниматься как очевидное: для каждого пользователя должно быть реализовано дешевое и удобное решение. Если системные администраторы не предоставят такую услугу, пользователи могут создать решение самостоятельно, которое будет менее безопасным и труднее управляемым. В случае возникновения проблем с самодельной службой от системных администраторов могут потребовать поддержку, что приведет к дополнительным сложностям в обслуживании.

Другие требования зависят от того, как пользователи собираются применять систему удаленного доступа. Наиболее частыми пользователями удаленного доступа являются люди, которые путешествуют и хотят проверять или отправлять электронную почту. Также популярны пользователи, которые нуждаются в доступе к системе вечером или в выходные, чтобы поработать несколько часов. Все эти пользователи требуют надежного и экономичного способа кратковременного подключения к сети. Однако они различаются в том, что одни ожидают возможности подключаться из любого места, а другие — только из дома.

Для пользователей, которые регулярно работают из дома, необходимо учесть три важных аспекта. Во-первых, система должна быть надежной, поскольку любые отключения будут негативно сказываться на их повседневной работе. Во-вторых, требуется высокоскоростное соединение, так как скорость соединения будет влиять на производительность, особенно при выполнении задач, требующих высокой пропускной способности или малой задержки. В-третьих, экономические факторы отличаются от периодического кратковременного использования: человек, работающий дома, может использовать удаленный доступ 8–10 часов в день, что делает оплату по времени нецелесообразной, и лучше выбрать безлимитное подключение.

В некоторых случаях сотрудники компании могут захотеть использовать высокоскоростное подключение к Интернету на конференции или дома. Эта ситуация особенно актуальна для технических компаний и требует использования шифрования для защиты конфиденциальной информации, которая передается через общедоступный Интернет. Без шифрования информация может быть перехвачена, а также может быть нарушен закон о защите интеллектуальной собственности. Шифрованное соединение должно обеспечивать доступ ко всем необходимым корпоративным ресурсам и быть достаточно производительным.

Служба удаленного доступа также должна удовлетворять потребности людей, работающих в другой компании, например, инженеров по поддержке или консультантов, которые работают в местоположении клиента. Этот случай добавляет сложность из-за необходимости учитывать политики безопасности двух несвязанных компаний. Обычно создание постоянного подключения между рабочим местом сотрудника и клиентской сетью неприемлемо, поскольку это может привести к нежелательной связи сетей. Часто решением является создание зашифрованного канала между машиной сотрудника и корпоративным брандмауэром через брандмауэр клиента и Интернет. Этот канал должен быть совместим с большинством брандмауэров и легко добавляться в сетевые конфигурации. Кроме того, механизм удаленного доступа должен быть гибким, обеспечивая сотруднику доступ ко всему, что ему необходимо в корпоративной сети. Однако это не всегда легко, так как некоторые приложения используют сложные протоколы, которые трудно пропустить через механизмы шифрования.

**27.1.2. Политика удаленного доступа**

Перед тем как начать предоставление удаленного доступа, компания должна определить свою политику в этой области. Политика должна включать в себя допустимые способы использования службы, требования безопасности, а также обязанности пользователей. В ней также должно быть указано, кто и какой тип удаленного доступа получает, а также кто оплачивает эти услуги. Подробнее об этом рассматривается в разделе 11.1.2, где также обсуждаются другие аспекты безопасности.

**27.1.3. Определение уровней обслуживания**

Очень важно четко определить уровни обслуживания для разных видов удаленного доступа, учитывая потребности пользователей. Удаленный доступ является чувствительной областью, так как сбои могут нарушить работу пользователей, которые не могут выполнить свою работу до восстановления службы. Если уровни обслуживания будут четко определены и пользователи осознают их, это поможет уменьшить уровень стресса и раздражения, поскольку они будут знать, сколько времени потребуется на устранение проблем.

Для небольших компаний с одним системным администратором уровни обслуживания должны быть такими, чтобы администратор мог отдыхать и эффективно работать в течение дня. В крупных компаниях с круглосуточной службой поддержки уровни обслуживания будут более ориентированы на пользователей. Персонал службы поддержки должен быть обучен работе с удаленным доступом и знать, когда передавать проблемы на более высокий уровень. В новых службах на пробном этапе уровни обслуживания обычно будут ниже, так как персонал еще не обучен и с технологией могут возникнуть проблемы. Также у пользователей пробных служб должен быть запасной метод доступа.

**27.1.4. Централизация**

Удаленный доступ выигрывает от централизации, как с точки зрения безопасности, так и с точки зрения расходов. Аутентификация и контроль доступа должны быть централизованы для обеспечения нормальной работы службы. С развитием новых технологий исследование и поддержка их реализации требуют значительных затрат, и их нецелесообразно дублировать в разных частях компании. Концентрация всех услуг на централизованном оборудовании и каналах связи позволяет достичь экономии за счет масштаба. В конечном итоге служба удаленного доступа может быть передана сторонним исполнителям.

**27.1.5. Привлечение сторонних исполнителей**

Наилучший способ справиться с постоянно меняющимися технологиями удаленного доступа – это передать эту задачу сторонним исполнителям. Постоянная оценка новых технологий и их адаптация требуют значительных усилий, а через несколько лет нужно будет перейти на более новые решения. Поэтому некоторые компании по предоставлению услуг удаленного доступа, такие как интернет-провайдеры, могут взять на себя часть этих задач. Они могут установить программное обеспечение VPN на устройствах пользователей, которые подключаются через обычные номера модемных пулов или другие способы, и обеспечивают соединение с корпоративной сетью через VPN.

Передача удаленного доступа сторонним исполнителям также позволяет значительно снизить время, затрачиваемое на его поддержку. Это делает расходы предсказуемыми и легко включаемыми в бюджет. Для системных администраторов это означает меньшее количество звонков по поддержке в нерабочее время и устранение необходимости поддерживать устаревшее оборудование, такое как модемы.

Однако не все аспекты удаленного доступа могут быть переданы сторонним исполнителям. Например, база данных аутентификации должна оставаться внутри компании для обработки увольнений сотрудников и обеспечения безопасности. В случае решения передать какие-либо функции сторонним исполнителям, системные администраторы должны тщательно выбрать поставщика услуг, чтобы избежать нежелательных последствий.

**27.1.6. Аутентификация**

Система аутентификации и авторизации является ключевым компонентом службы удаленного доступа. Даже если все другие аспекты предоставления удаленного доступа передаются сторонним исполнителям, систему аутентификации следует поддерживать внутри компании. Все методы удаленного доступа должны использовать единую базу данных аутентификации для минимизации затрат на администрирование и уменьшения вероятности ошибок, например, при отключении доступа уволенных сотрудников. Для доступа к этой базе данных можно использовать различные интерфейсы, в зависимости от требуемых протоколов.

Механизм аутентификации должен использовать безопасные методы, такие как одноразовые пароли или маркеры, а не пароли с многократным использованием. Для небольших компаний, возможно, начальная система будет основана на паролях, но важно предусмотреть переход к более безопасной системе в будущем.

**27.1.7. Безопасность периметра**

Служба удаленного доступа является частью периметра безопасности компании, даже если она передана сторонним исполнителям. Если компания полагается на защиту своего периметра, то важно, чтобы этот периметр оставался целостным. Неправильная настройка компонентов службы удаленного доступа, таких как узлы или сетевое оборудование, может привести к уязвимости, нарушая безопасность всей сети. Например, использование динамической маршрутизации в неправильных условиях может ослабить периметр безопасности, создавая новые угрозы.

**27.2. Тонкости**

Есть несколько способов улучшить службу удаленного доступа и сделать ее более эффективной, не увеличивая нагрузку на системных администраторов и не ухудшая производительность компании.

**27.2.1. Домашний офис**

Удаленный доступ – это только одна из частей решения для работы на дому. Кроме того, сотрудники, работающие удаленно, могут сталкиваться с различными проблемами, требующими решений на уровне оборудования и инфраструктуры. Один из основных вопросов касается того, кто обеспечивает оборудование для домашнего офиса и кто несет ответственность за его обслуживание. При этом стоимость предоставления поддержки для оборудования, которое требует установки или замены на дому, может быть высокой. SLA (соглашение об уровне обслуживания) должно четко регулировать такие вопросы, включая компенсацию затрат на оборудование и поддержку, если сотрудники работают из дома.

Кроме того, сотрудники могут столкнуться с необходимостью использования телефона, факса, принтера или участия в видеоконференциях, что также требует решения на уровне инфраструктуры. Компания может покрывать расходы на телефонные линии или другие услуги связи, чтобы минимизировать неудобства для сотрудников и избежать скрытых издержек на административные задачи (например, отчеты о расходах).

Для предотвращения социальной изоляции сотрудников, работающих удаленно, важно обеспечивать доступ к неформальному общению в компании, например, через видеоконференции и виртуальные встречи.

**27.2.2. Анализ и сокращение расходов**

Расходы на удаленный доступ могут быстро накапливаться и стать значительным бременем для компании. Системы удаленного доступа часто включают бесплатные номера телефонов для подключения, что может быть дорогостоящим для бизнеса. Чтобы снизить расходы, можно предоставить местные номера для областей с большим числом пользователей удаленного доступа. Также стоит рассмотреть возможность установки постоянных безлимитных соединений для сотрудников, которые используют услугу часто.

Максимальная автоматизация процесса подключения, включая уведомления для пользователей, что они могут использовать местный номер вместо бесплатного, может помочь уменьшить издержки без ухудшения качества обслуживания.

**27.2.3. Новые технологии**

Системные администраторы, поддерживающие службы удаленного доступа, сталкиваются с задачей не отставать от новых технологий, при этом необходимо поддерживать и старые решения. Это приводит к увеличению расходов на поддержку, поскольку одновременно приходится обслуживать несколько технологий. Одним из преимуществ традиционных модемов является обратная совместимость, что позволяет обновить инфраструктуру до более современных стандартов, поддерживая старое оборудование. Однако многие другие технологии не предлагают такого уровня гибкости.

Введение новых технологий увеличивает расходы на поддержку до тех пор, пока эта технология не станет популярной и не будет хорошо понята. С другой стороны, поддержка старых технологий тоже затратна, так как оборудование становится менее надежным, а системные администраторы теряют навыки работы с ним, особенно если технологией пользуется небольшое количество людей.

Для того чтобы контролировать расходы на поддержку, важно ограничить количество поддерживаемых технологий. Системным администраторам рекомендуется поддерживать не более двух новых технологий помимо традиционного модемного доступа. Когда приходит время перехода на новые решения, следует планировать вывод из эксплуатации старой технологии, переводя ее пользователей на более современные системы, с четким указанием даты, когда старое оборудование или решение будет исключено из работы.

**27.3. Заключение**

Поддержка службы удаленного доступа – это задача, которая требует значительных усилий и времени. Системные администраторы сталкиваются с проблемой необходимости постоянно обновлять технологии, чтобы удовлетворить потребности пользователей, но иногда ресурсы компании ограничены для реализации таких изменений.

Перед тем как приступать к построению службы удаленного доступа, важно понять все требования, которые могут быть разнообразными. Нужно разработать политику удаленного доступа, которая будет обязательной для пользователей, а также четко определить уровни обслуживания, чтобы пользователи знали, чего ожидать.

Централизация службы удаленного доступа помогает эффективно управлять изменениями технологий, а передача части обслуживания сторонним исполнителям может привести к экономии и улучшению качества обслуживания благодаря масштабируемости и большему покрытию. Однако важно сохранить контроль над элементами аутентификации и авторизации, а также уделять внимание безопасности службы.

Также важно заранее решить вопросы, которые могут возникнуть у сотрудников, работающих удаленно, такие как оборудование для домашнего офиса и вопросы связи. Автоматизация процессов и поиск способов снижения расходов помогут повысить эффективность службы и облегчить задачу системных администраторов.

**Глава 28. База программного обеспечения**

База программного обеспечения — это способ обеспечения доступности большого количества программных пакетов для множества узлов. В UNIX существует глобально доступная директория, например, /usr/local/bin, которая используется всеми узлами в кластере. В Windows же традиционно используются другие подходы, такие как хранилища для устанавливаемых дистрибутивов.

UNIX известна предоставлением множества средств для выполнения различных задач, однако это количество кажется ограниченным по сравнению с множеством средств, доступных через Интернет. Выбор и установка этих средств — важная ответственность. Без базы программного обеспечения пользователи могут столкнуться с нехваткой нужных инструментов, что снизит их производительность, а дублирование работы приведет к дополнительным затратам времени.

Хорошо настроенная база программного обеспечения делает даже слабую операционную систему полезной и функциональной. Это может стать привычкой для пользователей, но важно обеспечить доступность нужных инструментов для всех.

**Отличия между базами программного обеспечения в UNIX и Windows**

Исторически базы программного обеспечения для UNIX и Windows имеют значительные отличия. В UNIX базы обычно используются в реальном времени, а программы часто предоставляются в виде исходного кода. Установка таких программ может быть сложной и требовать прав root-доступа, чего обычно нет у пользователей. Поэтому в UNIX принято, чтобы одна группа людей или одна организация собирала и распространяла пакеты по всем узлам, часто синхронизируя десятки или сотни узлов. Это повышает эффективность обслуживания и упрощает поддержку.

Программы в UNIX часто не устанавливаются на локальной машине, а доступны на файловом сервере, к которому можно подключиться. Это позволяет централизованно обновлять программное обеспечение, а пользователи могут работать с программами напрямую с сервера.

В Windows подход отличается: базы программного обеспечения чаще всего включают установочные пакеты, которые сначала нужно установить, а затем настроить параметры, например, в реестре. Основные типы баз программного обеспечения для Windows включают:

1. **Сетевой диск**, где хранится программное обеспечение, и программы могут быть установлены на клиентской машине через сетевой доступ.
2. **Сетевые системы распространения программ** (например, Microsoft System Management Service - MS-SMS), позволяющие централизованно управлять установкой программ на множество машин.
3. **Модели сервера распространения**, где доступ к установочным файлам предоставляется через веб или локальную сеть.

**Политики и документация**

В этой главе будет рассмотрено обоснование для создания баз программного обеспечения, а также технические требования, которые следует учитывать при проектировании таких систем. Также будут рассмотрены политики, которые необходимо разработать, чтобы правильно управлять этими базами, и документация, которая должна быть составлена для правильной работы с базами.

Затем будет представлен процесс выбора программных решений для управления базами программного обеспечения и проектирование простых систем для UNIX и Windows. Система, которая будет спроектирована, будет легкой в использовании и потребует минимального программирования.

Эта глава позволяет понять, как эффективно организовать и управлять большими наборами программного обеспечения в различных операционных системах, с акцентом на автоматизацию, централизованную установку и синхронизацию.

**28.1.1. Обоснование**

База программного обеспечения представляет собой службу, которая находит, устанавливает и поддерживает библиотеку программ. Этот подход значительно снижает дублирование усилий, что, в свою очередь, способствует существенному снижению расходов. База программного обеспечения освобождает пользователей и администраторов от необходимости искать программы в Интернете или по каталогам, предлагая единое место для поиска и установки нужных программ.

Одним из основных преимуществ базы программного обеспечения является централизованный доступ, что исключает установку программ в разных местах сети. Когда программа появляется в базе, она сразу становится доступной для всех пользователей, обеспечивая целостность и согласованность среди всех машин в сети. Системные администраторы, занимающиеся поддержкой таких баз, активно отслеживают новые версии программ, исправления ошибок и обновления безопасности, что гарантирует, что все пользователи всегда имеют доступ к актуальной версии программного обеспечения.

Поддержка базы программного обеспечения требует от системных администраторов не только установки программ, но и систематического обновления базы, чтобы обеспечить стабильную работу всех программ и повысить производительность пользователей. Централизованный подход в значительной мере упрощает управление программами и их поддержание.

Для маленьких компаний идея использования базы программного обеспечения может казаться излишней, но с ростом компании, необходимость в централизованном хранении программных пакетов становится очевидной. Признаками того, что компания готова к использованию базы программного обеспечения, могут стать дублирование процессов установки и отсутствие организованной структуры хранения программ.

**28.1.2. Технические требования**

Проектирование базы программного обеспечения должно учитывать требования пользователей и обеспечивать функциональность, соответствующую их нуждам. Например, пользователи могут требовать, чтобы несколько версий одного инструмента, например компилятора, были доступны одновременно, особенно в переходный период на новые версии.

Особое внимание следует уделить надежности базы. Если программное обеспечение хранится локально на машинах, проблем с доступностью будет меньше, так как данные остаются доступными, пока работает сама машина. Однако если база размещена на удаленном сервере, необходимо учитывать надежность сети и сервера. В таких случаях можно применить репликацию или использовать методы RAID для повышения доступности и отказоустойчивости базы.

Также стоит оценить уровень доступности: для серверов, доступ к которым осуществляется редко, пользователи будут более снисходительно относиться к возможным сбоям. В случае, если база программного обеспечения критична для работы компании, следует уделить внимание надежности и масштабируемости серверной инфраструктуры для обеспечения бесперебойного доступа.

**28.1.3. Установите политику**

Необходима чёткая политика, которая регулирует, кто может размещать программные пакеты в базе, чтобы избежать рисков, таких как внедрение вредоносного ПО (например, троянских коней) или создание проблем с совместимостью из-за неправильных обновлений. Политика должна учитывать следующие моменты:

* **Кто может собирать и устанавливать пакеты?** Возможно, один или несколько человек, работающие исключительно с созданием пакетов, будут заниматься этим процессом. Для контроля качества и установки пакетов может быть назначен ответственный человек, даже если сборку могут выполнять все системные администраторы. Пользователи могут обновлять определённые пакеты, но первую установку должен контролировать системный администратор.
* **Что делать, если специалист покинет организацию?** Важно предусмотреть план действий на случай, если сотрудник, ответственный за поддержку базы, уходит. Это поможет избежать путаницы и сохранить эффективность работы базы.
* **Какие операционные системы поддерживаются?** Нужно определить, будет ли база для каждой операционной системы или использоваться единая база для разных систем, с учётом специфики. В UNIX можно иметь единую базу с использованием скриптов-оболочек для обработки различий между системами.
* **Стандартные оболочки и шаблоны.** Если для различных операционных систем используется одна база, важно иметь стандартные оболочки или шаблоны, которые должны применять все администраторы для настройки пакетов.
* **Как осуществляется обновление?** Важно определить, кто отвечает за установку обновлений, а также за установку новых версий пакетов. Это должен быть чётко прописанный процесс.
* **Как решаются ошибки?** Необходимо определить, кто отвечает за отладку пакетов с открытым исходным кодом: будут ли специалисты самостоятельно устранять ошибки или только собирать жалобы от пользователей и ждать исправлений в следующей версии.
* **Как пакеты удаляются из базы?** Для старых версий программ нужно предусмотреть строгую политику удаления. Пакеты могут удаляться, если они устарели или заменены более новыми версиями.
* **Охват распространения.** Нужно уточнить, будет ли база использоваться для всего кластера, подразделения или организации. Это определит масштаб и ресурсы, необходимые для поддержки базы.
* **Как пользователи могут запрашивать программы?** Важно, чтобы существовал механизм подачи запросов на добавление программ в базу. Может быть создана комиссия, которая будет решать такие вопросы.
* **Публикация политики.** Политика должна быть доступна всем пользователям и быть размещена в удобном для них месте, например, на внутреннем портале или в документировании компании.

**Отслеживание лицензий**

Для отслеживания лицензий программного обеспечения рекомендуется использовать простые, но эффективные методы, например, создание базы данных с учётом всех лицензионных требований программ. В разделе 12.1.5 описаны некоторые способы отслеживания, которые помогут избежать нарушений.

**28.1.4. Выбор программы для базы**

Для UNIX-систем рекомендуется выбрать существующую систему управления базой программного обеспечения, а не разрабатывать её с нуля. Важно понимать, что гораздо проще настроить уже существующие решения, чем разрабатывать новые, даже если ваша среда имеет специфические требования. Существуют бесплатные пакеты, такие как:

* **Depot** (Colyer and Wong 1992) — классическая программа для администрирования баз программ.
* **LUDE** (Dagenais et al. 1993), **Modules** (Furlani and Osel 1996), **SEPP** (Oetiker 1998b) — другие программы для управления пакетами.
* **GNU Stow** (Glickstein 1996) — полезное средство для управления символическими ссылками в базе UNIX.

Для Windows также существуют варианты. Например, можно использовать FTP-сервер или веб-сайт для хранения и распространения программ. В разделе 28.1.6.2 рассматривается, как можно организовать и документировать базу программ для Windows.

Если в компании уже используется устаревшая система хранения программ, следует решить, будет ли целесообразно её усовершенствование или создание новой.

**28.1.5. Создайте руководство для процесса**

Для эффективной работы с базой программного обеспечения важно создать детализированное руководство, которое описывает процесс добавления новых программных пакетов. Хотя документация, предоставляемая с диспетчером пакетов, может быть полезной, важно учитывать локальные особенности, такие как имена узлов и уникальные настройки системы.

После того как люди научатся добавлять несколько пакетов, будет полезно создать краткое руководство, которое подытожит шаги без лишних объяснений, чтобы опытные системные администраторы могли быстро выполнить процесс. Это руководство должно быть достаточно простым, чтобы можно было скопировать и вставить команды в командную строку, ускоряя выполнение задач.

**28.1.6. Примеры**

**28.1.6.1. UNIX**

Пример базы программного обеспечения для UNIX, которая может использоваться как на одной системе, так и в распределенной сети, приведен ниже. Система поддерживает установку различных версий языка программирования Perl и почтового клиента mutt.

1. **Структура исходного кода**:
   * Исходный код каждого пакета хранится в директории /home/scr, с поддиректориями для каждого пакета. Например, для Perl будут файлы perl-6.0.tar.gz и perl-6.1.tar.gz, а также исходные коды для сборки. В случае нескольких ОС могут быть дополнительные уровни поддиректорий для различных платформ.
   * В директории /home/scr также хранится скрипт SOURCEME, который настраивает параметры среды для правильной сборки пакета. Это облегчает сборку новых версий пакетов.
2. **Установка пакетов**:
   * Пакеты устанавливаются в директории, именуемые по имени пакета и версии, например, /sw/perl-6.0 для версии 6.0 Perl, или /sw/mutt-2.0.1 для версии mutt. Это позволяет администраторам одновременно поддерживать несколько версий программы.
   * Название пакета без версии (/sw/perl) является символической ссылкой на последнюю поддерживаемую версию. Например, /sw/perl будет ссылаться на /sw/perl-6.1.
3. **Обновление пакетов**:
   * Когда появляется новая версия программы, например Perl 7.0, она устанавливается в директорию /sw/perl-7.0. Пользователи могут тестировать новую версию, добавляя в свой PATH путь к новой директории, но основной символической ссылкой для всех пользователей будет являться версия 6.1, пока новая не пройдет тестирование.
   * При установке новой версии важно, чтобы старая версия продолжала работать, не вызывая проблем у процессов, которые её используют.
4. **Управление длинными путями**:
   * Чтобы избежать трудностей с управлением длинной переменной PATH, создаётся специальная директория /sw/default/bin, которая содержит символические ссылки на самые популярные программы. Все пользователи могут добавить эту директорию в свой PATH, чтобы быстро получить доступ к общим программам без необходимости включать в PATH каждую версию отдельно.
5. **Автоматизация создания ссылок**:
   * Для управления символическими ссылками можно использовать программу GNU Stow, которая упрощает процесс создания ссылок на двоичные файлы, библиотеки и страницы man. Это позволяет избежать ручного создания ссылок для каждого файла в пакете.
6. **Обеспечение доступа в сети**:
   * Для обеспечения доступа к программному обеспечению на нескольких узлах сети, пакеты могут быть скопированы на соответствующие машины или доступ к ним может быть организован через NFS (сетевая файловая система). В таком случае все узлы могут использовать одну и ту же версию пакета.

Таким образом, эта система позволяет эффективно управлять и поддерживать пакеты на нескольких узлах с возможностью обновления, тестирования и использования нескольких версий одновременно, минимизируя сложность и риски для пользователей.

**28.1.6.2. Windows**

В средах Windows узлы традиционно администрируются более независимо, чем в UNIX-средах. В отличие от централизованного подхода UNIX, где пакеты хранятся на сервере и распространяются по сети, в Windows хранилище программного обеспечения чаще напоминает FTP-сервер или файловый сервер, где хранится набор установочных пакетов, таких как ZIP-файлы или автоматические установщики в формате .EXE. Эти пакеты могут быть установлены пользователями на их собственных компьютерах.

Политики, касающиеся установки программного обеспечения, могут сильно различаться. В некоторых организациях полностью запрещена установка программ, в других разрешено устанавливать только одобренные программы, а в третьих пользователи могут устанавливать любые программы. Важно, чтобы структура базы программного обеспечения соответствовала этим политикам.

Пример простой базы программного обеспечения для Windows:

1. **Основная структура папок**:
   * **software**: Основная папка, доступная по пути \\server1\software, где пользователи могут искать и устанавливать программы, доступные для их систем.
     + **POLICY**: Документ, объясняющий политику установки программ, ограничения, связанные с отдельными пакетами.
     + Подкатегории:
       - **Стандартные**: Программы, одобренные для установки на все машины, но не предустановленные. Здесь могут быть размещены программы с корпоративными лицензиями.
       - **Предустановленные**: Программы, которые должны быть установлены по умолчанию или автоматически обновляться через MS-SMS. Это могут быть программы типа WinZip, антивирусные приложения, офисные пакеты.
       - **Образы дисков**: Образы CD/DVD, например, для дистрибутивов Linux или BSD. Это позволяет сократить трафик интернет-шлюза.
       - **Экспериментальные**: Программы, которые находятся в процессе тестирования и одобрения. Доступ ограничен паролем или через ACL (список контроля доступа).
       - **Администратор**: Программы, доступ к которым есть только у системных администраторов, включая программы с индивидуальными лицензиями. Эти папки защищены паролем или ACL.
2. **Структура каждой папки**: Каждая папка может содержать дополнительные подпапки для различных версий Windows, например, Windows 10 или Windows 7. Программы, которые больше не поддерживаются, можно перемещать в папку **Устаревшие**.
3. **Хранение пакетов**: Пакеты хранятся в отдельных папках с понятными именами, например, FooSoft Accounting Client 4.0. В каждой папке должна быть информация о программе, лицензировании и инструкции по установке. Это помогает пользователям и администраторам быстро ориентироваться в содержимом.
4. **Документация**: Для каждого пакета создается отдельный документ с подробной информацией о нем и особенностях установки, чтобы избежать путаницы и гарантировать корректную установку программ.
5. **Репликация базы программ**: Если база программного обеспечения будет успешной, можно подумать о ее распространении по нескольким частям компании для снижения нагрузки на сеть и повышения скорости установки. Репликация базы данных программ под Windows не столь критична с точки зрения реального времени, как в UNIX, где необходим постоянный доступ к сети. В Windows система установки не зависит от постоянного сетевого доступа после начальной установки программы.

Эта структура представляет собой мощную, но гибкую базу программного обеспечения, которая может быть легко адаптирована для нужд различных организаций, учитывая особенности администрирования в среде Windows.

### 28.2. Тонкости

Разработка и управление базой программного обеспечения требует учета нескольких специфических потребностей. Часто необходимо обеспечить индивидуальные настройки для разных узлов, локальную репликацию пакетов, работу с коммерческими программами и меньшие базы для ОС, которые не получают полной поддержки.

#### 28.2.1. Различные конфигурации для разных узлов

Одной из часто требуемых функций баз программного обеспечения в UNIX является возможность обеспечения различных конфигураций для разных узлов или кластеров. Это может быть полезно, если конфигурация пакета должна сильно отличаться на разных узлах.

* **Пример использования символических ссылок**:
  + Файл конфигурации может быть символической ссылкой на локальный файл. Например, /sw/megasoft/lib/megasoft.conf может ссылаться на /etc/megasoft.conf, который имеет особое содержимое для конкретного узла.
  + Если нужно выбирать между несколькими конфигурациями, можно включить их в пакеты. Например, /etc/megasoft.conf может быть символической ссылкой на один из файлов конфигурации, таких как megasoft.conf-server или megasoft.conf-client.

Использование символических ссылок позволяет гибко управлять конфигурациями, адаптируя их под нужды различных узлов.

#### 28.2.2. Локальная репликация

Если доступ к базе программ осуществляется по сети, может быть полезно локально хранить часто используемые пакеты, чтобы уменьшить нагрузку на сеть и повысить производительность.

* **Преимущества локальной репликации**:
  + Снижение сетевой нагрузки и повышение скорости работы.
  + Возможность кэширования программ на рабочих станциях с достаточным свободным местом.

Однако важно учесть, что управление локальными копиями может быть сложным, особенно когда нужно отслеживать версии пакетов и обновления. Программы управления базами должны упростить этот процесс, предоставляя статистику и позволяя системным администраторам или пользователям управлять кэшированием пакетов.

* **Примечание**: Локальная репликация может повлиять на систему резервного копирования. Нужно либо избегать резервного копирования локальных копий, либо продумать, как учесть их при резервном копировании и восстановлении.

#### 28.2.3. Коммерческие программы в базе

Включение коммерческих программ в базу программного обеспечения связано с рядом проблем, особенно когда речь идет о лицензировании.

* **Если программа имеет корпоративную лицензию**:
  + Она может быть включена в базу и использоваться на всех узлах. Если программа подключается к серверу лицензий для проверки доступности, она будет бесполезна для неавторизованных пользователей.
* **Проблемы с лицензиями для конкретных пользователей**:
  + Если программа доступна только для определенной группы пользователей, то администратор должен обеспечить выполнение условий лицензии. Это может включать установку программы только на узлы, которые авторизованы для ее использования.
* **Решение для UNIX-среды**:
  + Можно использовать группы пользователей, которые имеют доступ к лицензированным программам, и ограничивать доступ к ключевым файлам в пакете только для этих групп.

В некоторых случаях, например, если различные группы используют одну программу с собственным сервером лицензий, можно использовать перенаправляющие ссылки для указания разных файлов лицензий для разных клиентов.

### Резюме:

* **Конфигурации для разных узлов**: Использование символических ссылок для адаптации конфигураций под различные узлы.
* **Локальная репликация**: Повышение производительности за счет кэширования часто используемых пакетов, но с учетом сложности управления обновлениями.
* **Коммерческие программы**: Необходимость особого контроля доступа и лицензирования для коммерческих программ в базе, использование групп и перенаправляющих ссылок для управления доступом.

### 28.2.4. Граждане второго сорта

В любой сети могут быть узлы, работающие под операционными системами, которые не получают полной поддержки, как системы «граждан первого сорта». Такие узлы часто называют «гражданами второго сорта». Эти машины могут требовать минимальной настройки и ограниченной поддержки, например, просто выделения IP-адреса и настройки базовых параметров конфигурации, чтобы устройства работали.

#### Поддержка граждан второго сорта

Для таких узлов база программного обеспечения обычно минимальна и включает только необходимые для них приложения. Это может быть компилятор или специализированное программное обеспечение, требуемое для выполнения специфических задач машины, а также инструменты для работы системных администраторов.

* **Что не нужно делать**: Не стоит пытаться предоставлять все пакеты, доступные для полноценно поддерживаемых систем. Это будет требовать больших усилий при малой отдаче.

#### Политика поддержки граждан второго сорта

Важно иметь письменную политику, которая описывает уровень поддержки для таких узлов и определяет, какие задачи ложатся на системного администратора, а какие — на пользователей. Эта политика должна содержать:

1. Минимальный набор пакетов, доступных для этих узлов.
2. Уровень поддержки, на который пользователи могут рассчитывать.
3. Области, в которых пользователи должны самостоятельно обеспечивать поддержку.

#### Рекомендуемые инструменты для граждан второго сорта

Для узлов, работающих с ОС граждан второго сорта, следует устанавливать несколько групп инструментов:

1. **Инструменты для выполнения задач машины**:
   * Например, для переноса программ на эту ОС — соответствующие инструменты компилятора.
   * Для запуска конкретного приложения или службы — необходимые для этого программы.
2. **Средства для системного администрирования**:
   * Программы для сбора информации об оборудовании, логирования, резервного копирования, обновлений базы программного обеспечения, отладки и т. д.
3. **Полезные утилиты для компании**:
   * Например, консольная версия корпоративного телефонного справочника, минимальные конфигурации электронной почты (например, для отправки).

#### Цепочка инструментов

Цепочка инструментов разработчика представляет собой набор программ, которые необходимы для разработки программного обеспечения. Эта цепочка может включать:

* Средства сборки (например, make, autoconf),
* Компиляторы, ассемблеры, компоновщики,
* Интерпретаторы и отладчики,
* Системы контроля версий (например, SubVersion, CVS, Perforce, ClearCase),
* Различные утилиты, используемые в процессе разработки.

Термин «цепочка» подразумевает, что инструменты передают данные друг другу в процессе разработки, как, например, в последовательности компиляции, ассемблирования и компоновки.

### 28.3. Заключение

В этой главе рассматривались базы программного обеспечения, которые представляют собой организованный способ распространения программных пакетов на большое количество узлов, хотя они полезны и для одного узла. Хорошо организованная база программного обеспечения предоставляет полный набор инструментов и становится важной частью культуры пользователей и самой сети.

* **Отличие баз для Windows и UNIX**: В Windows базы программного обеспечения обычно используются для установки программ, в то время как в UNIX — для использования программ в реальном времени.
* **Политика для базы программного обеспечения**: В каждой организации должна быть письменная политика, описывающая, как и кем устанавливаются программы, какие системы обслуживаются базой, как выполняются запросы и поддержка.
* **Рекомендации**: Для управления базами программного обеспечения рекомендуется использовать существующие пакеты с открытым исходным кодом, так как создание базы с нуля может быть сложным и затратным. Найдите подходящий инструмент и адаптируйте его под свои нужды.

### Глава 29: Веб-службы

Управление веб-системами стало важной частью системного администрирования, и с развитием технологий веб-службы требуют все более специализированных подходов и квалифицированных специалистов. Хотя теоретически многие вопросы, связанные с запуском веб-службы, уже были рассмотрены, в этой главе мы подробно остановимся на важных аспектах, которые не были затронуты в предыдущих разделах.

### Веб-сайт и веб-приложения

Веб-сайт представляет собой средство представления информации и приложений в модели клиент/сервер. Веб-контент, как правило, доступен через браузер, но доступ возможен и через другие программы, поддерживающие HTTP. Веб-серверы отвечают за предоставление документов в формате HTML и других типов контента, таких как графика, аудио и видео файлы. Также веб-страницы могут включать программы, например, на JavaScript, которые выполняются в браузере пользователя.

Веб-система основана на открытых стандартах, которые позволяют различным программам и устройствам взаимодействовать друг с другом без необходимости платить за лицензии или собственность. Веб-стандарты разрабатываются консорциумом W3C, а сетевые протоколы — IETF.

### Преимущества веб-приложений

Одним из главных преимуществ веб-приложений является универсальность. Один веб-браузер может использоваться для доступа ко множеству разных веб-приложений. Это позволяет создавать гибкие решения, которые могут быть использованы на различных устройствах, от персональных компьютеров до мобильных телефонов.

Кроме того, многие устройства, такие как маршрутизаторы, коммутаторы и интеллектуальные дисковые массивы, имеют встроенные веб-серверы и веб-приложения, что позволяет управлять ими через браузер.

### Открытые стандарты

Если бы компания разработала собственные стандарты и взимала плату за их использование, интернет, вероятно, не был бы таким развитым, как сейчас. Такой подход мог бы привести к созданию закрытых экосистем, как это произошло с продуктами Network Notes от AT&T, Lotus и Novell. Это была попытка создания изолированной сети, доступной только для пользователей, использующих соответствующие программы. Однако из-за ограничений в доступе и отсутствии совместимости с другими системами эта идея не получила популярности.

**Пример Network Notes**: Компаниям пришлось бы покупать лицензии на программы, и все участники сети были бы ограничены использованием только этих программ. Это привело бы к отсутствию открытых решений, и, следовательно, разработчики не смогли бы экспериментировать с созданием альтернативных программ, например, веб-браузеров для мобильных телефонов.

Хотя в истории интернета уже были попытки создать закрытые стандарты, такие как в Network Notes, они потерпели неудачу, и в итоге веб стал успешным благодаря открытым стандартам. Однако, несмотря на успех интернета, современные компании иногда пытаются ограничить возможности пользователей с помощью закрытых решений, что препятствует инновациям и удобству пользователей.

### 29.1. Основы

В этом разделе будут рассмотрены основные элементы, составляющие веб-службы, распространенные архитектуры, а также необходимые меры для создания безопасной, расширяемой и наблюдаемой веб-службы. Важность этих аспектов заключается в необходимости обеспечения стабильности работы веб-приложений, а также в поддержке их масштабируемости и безопасности.

Веб-службы должны быть спроектированы так, чтобы их можно было легко управлять, масштабировать и обновлять, учитывая требования безопасности и наблюдаемости для эффективного мониторинга их состояния и устранения возможных проблем.

### 29.1.1. Основные элементы веб-службы

Веб-службы, как и любые другие сетевые сервисы, состоят из нескольких ключевых компонентов, которые позволяют эффективно передавать данные между сервером и клиентом. Рассмотрим основные элементы, которые составляют эти службы.

#### Унифицированный указатель ресурса (URL)

URL (Uniform Resource Locator) — это адрес ресурса в Интернете. Он указывает на расположение информации на сервере. URL состоит из двух основных частей:

* **Имя узла** (например, www.EverythingSysadmin.com)
* **Путь к ресурсу** (например, / для домашней страницы сайта).

#### Веб-серверы

Веб-сервер получает запросы от клиентов (броузеров) и возвращает ответ в виде данных. Наиболее распространенные веб-серверы:

* **Apache HTTP**,
* **AOLServer**,
* **Microsoft IIS**.

Ответы сервера обычно бывают **статическими**, например, HTML-страницы. Однако иногда сервер генерирует **динамическое** содержимое, которое зависит от запроса пользователя, например, с помощью **CGI** (Common Gateway Interface).

#### CGI и динамическое содержимое

С помощью CGI веб-сервер может генерировать динамическое содержимое, которое часто включает запросы к базе данных. CGI-скрипты могут быть двух типов:

* **GET**: используется для получения данных (например, для поиска на сайте). В запросе передаются данные через URL.
* **POST**: используется для отправки данных, например, для обновления корзины в интернет-магазине или отправки сообщений.

Важно отметить, что **GET** используется для чтения данных, а **POST** — для их изменения. Ошибки, такие как использование метода GET для удаления данных, могут привести к непреднамеренным последствиям, например, удалению данных при сканировании поисковыми роботами.

#### Внутренние интерпретируемые коды

Кроме CGI, существуют другие технологии для динамического создания веб-страниц, такие как:

* **PHP**,
* **Microsoft Active Server Pages (ASP)**.

Эти технологии позволяют серверу встраивать код в HTML-страницы, который выполняется до того, как страница будет отправлена пользователю.

#### Веб-клиенты

**Веб-клиенты** — это программы, которые взаимодействуют с веб-серверами. Наиболее распространенный веб-клиент — это **веб-браузер**. Браузер выполняет несколько функций:

* Интерпретирует и отображает HTML-страницы.
* Выполняет встроенные скрипты, такие как **JavaScript**.
* Обрабатывает другие форматы данных, например изображения, видео и аудио.

С помощью веб-браузеров пользователи могут получать доступ к веб-сайтам и взаимодействовать с ними, не устанавливая дополнительных программ. Это значительно ускоряет развитие веб-технологий.

#### AJAX и интерактивные страницы

**AJAX** (Asynchronous JavaScript and XML) — это метод создания интерактивных веб-страниц, который позволяет обмениваться данными с сервером асинхронно, не обновляя всю страницу. В результате, страницы могут быть такими же интерактивными, как и традиционные настольные приложения. В AJAX основной интерфейс создан с использованием JavaScript, который позволяет отправлять запросы к серверу без необходимости ожидания действия пользователя, как это происходит в стандартных HTTP-запросах (GET и POST).

#### Форматы данных и мультимедийные файлы

Веб-серверы часто используют различные форматы данных:

* **HTML** или его производные (XHTML, DHTML, XML),
* **Мультимедийные форматы** для изображений, видео и аудио.

При работе с мультимедийными файлами существует два подхода:

1. **Потоковая передача данных**: для аудио и видео в реальном времени, когда данные начинают воспроизводиться до того, как загрузка будет завершена (например, онлайн-радио).
2. **Загрузка всех данных до начала воспроизведения**: используется для статических файлов, таких как изображения.

#### XML и микроформаты

**XML** является универсальным форматом для передачи данных и создания других форматов. Он также широко используется для разработки микроформатов, таких как **RSS**. RSS-каналы позволяют отображать актуальное содержимое с новыми данными, например, новости или обновления блога.

Таким образом, веб-службы включают в себя множество различных технологий и компонентов, которые работают вместе, чтобы обеспечить взаимодействие между пользователями и веб-ресурсами. Важно учитывать, что с развитием мобильных устройств и технологий IoT, веб-клиенты становятся все более универсальными, а веб-сервисы — более адаптивными и динамичными.

### 29.1.2. Роль веб-мастера

**Веб-мастер** — это специалист, который управляет содержимым веб-сайта, подобно тому, как редактор управляет содержимым газеты. Его основная ответственность заключается в установке политики веб-сайта, обновлении и поддержке информации на сайте. Важно отметить, что эта роль отличается от роли **системного администратора веб-сервера**, который отвечает за техническую часть, включая установку серверного ПО, обеспечение функционирования серверов и устройств, которые поддерживают веб-сайт.

#### Различие между веб-мастером и системным администратором

* **Веб-мастер** управляет контентом и функциональностью сайта.
* **Системный администратор** отвечает за технические аспекты работы веб-сервера.

В малых компаниях один человек может выполнять обе функции, но в крупных компаниях эти роли должны быть разделены для повышения эффективности работы.

#### Рекомендации для системных администраторов

Если системный администратор вынужден выполнять обязанности веб-мастера, важно организовать структуру, которая позволит пользователям самостоятельно обновлять контент. Это поможет избежать непредвиденных ситуаций, когда нужно вносить изменения в последний момент, например, в канун выходных.

Системные администраторы могут избежать таких проблем, создав систему для **самообслуживания** веб-сайта, которая позволяет пользователям вносить изменения без необходимости вовлечения технического специалиста. Также рекомендуется устанавливать **SLA (соглашения об уровне обслуживания)**, в которых будет прописано время, за которое должны быть запрашиваемы изменения, и четкий график обновлений.

### 29.1.3. Соглашения об уровне обслуживания (SLA)

Для веб-службы, как и для любой другой службы, необходимо разработать **SLA** и системы мониторинга для контроля соблюдения этих соглашений. SLA для веб-службы может сильно варьироваться в зависимости от ее назначения и критичности.

#### Основные моменты SLA для веб-службы:

* SLA должно включать показатели задержки обработки запросов, например, **QPS (Queries Per Second)**, который определяет время, необходимое для обработки запросов при определенной нагрузке.
* Важно предусмотреть возможность проведения планового обслуживания, особенно если инфраструктура не избыточна.
* Для внутренних веб-служб SLA обычно будет меньше по жесткости, чем для публичных сервисов, доступных 24/7.

#### Пример соглашения:

* **Высокий уровень доступности**: если веб-служба критична и должна быть доступна круглосуточно, следует обсудить внедрение избыточности.
* **Мониторинг и управление**: SLA также включает в себя системы мониторинга, которые помогут отслеживать и регулировать задержки и производительность веб-службы.

### 29.1.4. Архитектуры веб-служб

Разные типы содержимого требуют различных инфраструктур подачи и обработки данных. Основные виды архитектуры веб-служб:

#### 29.1.4.1. Статический веб-сервер

**Статический веб-сервер** предоставляет неизменяемые или редко изменяющиеся документы (например, HTML-страницы, изображения, текстовые файлы). Эти документы обычно считываются с диска и не изменяются в процессе доставки.

* Для обеспечения быстрой работы сервера с высокой нагрузкой, важно, чтобы сервер обладал достаточным количеством оперативной памяти для кэширования наиболее часто запрашиваемых файлов.
* Статический веб-сервер легко настраивается и может работать с различными правами доступа для разных групп пользователей.

#### 29.1.4.2. CGI-серверы

**CGI-серверы** (Common Gateway Interface) генерируют динамическое содержимое на основе запросов пользователей. В отличие от статических серверов, CGI-серверы выполняют множество операций для создания каждой страницы, что требует больше ресурсов, особенно процессора.

* Для работы с высокими уровнями нагрузки (QPS) CGI-серверу нужно достаточное количество процессорных мощностей, а также оптимизация использования оперативной памяти и сети.
* CGI-серверы лучше подходят для динамически изменяющихся страниц, таких как результаты поиска или страницы с персонализированным содержимым.

Таким образом, архитектура веб-службы должна быть адаптирована к типу контента, который она обрабатывает, и уровню нагрузки, с которым она сталкивается. Важно также учитывать безопасность, особенно для публичных веб-сервисов, доступных через Интернет.

### 29.1.4.3. Веб-сайты на основе баз данных

Одним из наиболее распространенных типов веб-приложений является **веб-сайт на основе базы данных**, который позволяет пользователям взаимодействовать с информацией в базе данных, включая её обновление и модификацию. Примеры таких приложений включают социальные сети, интернет-магазины и образовательные платформы.

#### Преимущества веб-сайтов на основе базы данных:

1. **Гибкость и масштабируемость**: Вместо создания отдельной веб-страницы для каждого элемента, информация хранится в базе данных, и страницы формируются динамически. Например, в интернет-магазине каталог товаров записан в базе данных, а конкретная страница товара генерируется при обращении к шаблону с данными из базы.
2. **Управляемость**: Использование базы данных позволяет легко обновлять и управлять большим количеством информации. Например, при изменении шаблона можно глобально обновить формат отображения товаров или других данных на сайте.
3. **Упрощение администрирования**: В отличие от статических сайтов, где нужно вручную редактировать каждую страницу, сайты на базе данных автоматически обновляют содержимое, что упрощает администрирование.
4. **Масштабируемость и производительность**: Для сайтов с высокой нагрузкой (высокий QPS) используются стандартные методы масштабирования и повышения производительности баз данных, такие как репликация, кэширование и распределение нагрузки.

#### Проблемы и рекомендации:

Веб-сайты, которые не перешли на модель базы данных и всё ещё используют ручное редактирование страниц, становятся трудными для управления. Важно как можно скорее перевести такие сайты на систему с базой данных.

### 29.1.4.4. Мультимедийные серверы

**Мультимедийный сервер** — это веб-сервер, который предоставляет медиа-файлы, такие как видео или аудио. Эти файлы могут быть большими по объему и требуют специализированных решений для их хранения и передачи.

#### Особенности мультимедийных серверов:

1. **Хранение и пропускная способность сети**: Мультимедийные файлы занимают много места, а их передача требует высокой пропускной способности сети. Важно иметь систему хранения, оптимизированную для работы с большими, редко обновляемыми файлами.
2. **Потоковая передача**: Мультимедийные серверы часто предоставляют поддержку потоковой передачи, что позволяет пользователю просматривать или слушать контент в реальном времени, без необходимости загрузки всего файла. Это достигается с помощью специальных протоколов, которые позволяют передавать данные по частям.
3. **Буферизация и контроль скорости**: Чтобы обеспечить плавную передачу контента, сервер обычно выполняет **буферизацию**, предзагружая несколько секунд данных. Это позволяет компенсировать задержки в сети и обеспечить непрерывный поток.
4. **Производительность**:
   * Потоковые серверы могут эффективно использовать память и сеть, загружая в память лишь необходимые части медиа-файла. Это особенно эффективно для нескольких пользователей, которые начинают просмотр в разное время, так как данные могут быть повторно использованы.
   * Важно также учитывать **нагрузку на процессор**: при потоковой передаче медиа-контента может потребоваться перекодировка файлов в зависимости от устройства пользователя. Это требует значительных вычислительных ресурсов, особенно если файлы имеют высокое качество. Для снижения нагрузки часто используются специализированные аппаратные решения.

#### Проблемы и рекомендации:

Для мультимедийных серверов, предоставляющих множество одновременных потоков, необходимо тщательно планировать **пропускную способность сети** и **мощности системы хранения**. Кроме того, важно оптимизировать использование **системных ресурсов** (памяти, процессора и сети), чтобы обеспечить стабильную работу при высоких нагрузках.

В целом, мультимедийные серверы требуют особого внимания к инфраструктуре и архитектуре, чтобы эффективно управлять потоками данных и обеспечивать высококачественный пользовательский опыт.

### 29.1.4.5. Несколько серверов на одном узле

Существует два основных способа размещения нескольких серверов на одном физическом узле (сервере) без необходимости использования отдельной машины для каждого веб-сайта:

1. **Использование различных портов**:
   * Один сервер может быть настроен для работы на разных портах. Например, веб-сервер может слушать не стандартный порт 80, а порт 8001: http://my.website:8001/.
   * Это позволяет запускать несколько веб-серверов на одном узле, каждый из которых обслуживает свой сайт или службу, не требуя отдельного физического сервера.
   * **Преимущества**: Упрощение администрирования, минимизация привилегированного доступа для персонала.
   * **Проблемы**: Пользователи могут забывать указывать номер порта, что приведет к путанице, так как они могут попасть на другой сайт.
2. **Использование нескольких виртуальных сетевых интерфейсов (VIF)**:
   * В этом случае, несколько IP-адресов назначаются одному физическому сетевому интерфейсу через создание виртуальных интерфейсов (VIF). Это позволяет каждому веб-серверу работать с уникальным IP-адресом, несмотря на то, что они физически находятся на одном сервере.
   * **Преимущества**: Легкость в расширении, так как каждый сайт или служба настроены отдельно и можно быстро перенести их на новый сервер, просто изменив IP-адрес.
   * **Недостатки**: Требует дополнительной настройки, но это может быть выгодным в долгосрочной перспективе, особенно для масштабируемых решений.

Эти методы могут быть полезны для администрирования большого количества веб-сайтов без необходимости расширения инфраструктуры на дополнительные физические серверы.

### 29.1.5. Мониторинг

**Мониторинг веб-служб** — ключевая часть управления производительностью и безопасностью веб-сайтов. Он помогает отслеживать, как хорошо работают ваши сервисы, и позволяет выявлять проблемы на ранних стадиях.

#### Важные аспекты мониторинга:

1. **Ошибки веб-серверов**: Наиболее часто ошибки связаны с контентом сайта, такими как проблемы с загрузкой страниц или неправильные ссылки. Они могут быть важными для команды разработчиков, поскольку могут указывать на проблемы с кодом сайта или его скриптами.
2. **Поддержка расширенного мониторинга**: Многие веб-серверы поддерживают функции мониторинга, которые могут записывать информацию о браузере пользователя, URL страницы, которая привела на ваш сайт, а также данные о текущем потоке запросов и использовании памяти.
3. **Мониторинг безопасности**: Важно отслеживать попытки вторжений или подозрительные запросы. Наблюдение за ошибками и трендами может помочь быстро обнаружить угрозы безопасности.
4. **Использование платформы веб-сервера**: Обратитесь к документации веб-сервера для настройки и использования всех доступных функций мониторинга. Некоторые платформы предоставляют расширенные инструменты, которые позволяют более эффективно отслеживать производительность и ошибки.

### 29.1.6. Расширение веб-служб

**Расширение** (scaling) веб-служб — это критический процесс, который позволяет обеспечивать стабильную работу при увеличении количества пользователей и запросов.

#### Основные типы расширения:

1. **Горизонтальное расширение**:
   * Добавление новых серверов (машин) для распределения нагрузки. Это может быть достигнуто через использование кластеров или балансировщиков нагрузки.
   * Преимущества: Легкость в масштабировании, добавление серверов по мере необходимости.
   * Недостатки: Требует сложной инфраструктуры и настройки для обеспечения бесперебойной работы.
2. **Вертикальное расширение**:
   * Увеличение мощности одного сервера (улучшение процессора, добавление памяти, увеличение объема хранения данных).
   * Преимущества: Быстрое улучшение производительности при недостаточности ресурсов.
   * Недостатки: Ограничено возможностями одного устройства, трудности при дальнейшей масштабируемости.

#### "Slashdot-эффект":

Это явление, когда сайт, упомянутый на популярном новостном сайте Slashdot, неожиданно сталкивается с перегрузкой из-за резкого увеличения трафика. Это подчеркивает важность своевременного планирования масштабируемости, чтобы обеспечить бесперебойную работу при пиковых нагрузках.

### 29.1.6.1. Горизонтальное расширение

**Горизонтальное расширение** заключается в добавлении дополнительных серверов для распределения нагрузки между ними. Это позволяет улучшить производительность и надежность веб-сервиса.

#### Примеры и подходы:

1. **Циклические записи DNS**:
   * Один из способов распределить нагрузку — это использование циклической записи DNS, при которой запросы к одному доменному имени (например, [www.example.com](http://www.example.com/)) направляются на различные IP-адреса.
   * Однако, этот метод имеет недостатки, такие как кэширование IP-адресов в операционных системах или браузерах, что может привести к неправильному распределению нагрузки. Также могут возникать проблемы при сбое сервера, так как DNS может продолжать предоставлять адрес неработающего сервера.
2. **Балансировка нагрузки с помощью аппаратных устройств**:
   * Балансировщики нагрузки устанавливаются между пользователем и серверами, направляя запросы на различные сервера в зависимости от нагрузки или состояния серверов.
   * Такие устройства следят за состоянием серверов и прекращают направлять запросы на неработающие серверы, что повышает надежность системы. Также балансировщики могут направлять запросы на менее загруженные серверы.
   * Они также поддерживают различные протоколы, не только HTTP, что добавляет гибкости в архитектуру.

#### Проблемы с балансировщиками нагрузки:

В одном из примеров, система, использовавшая балансировщики нагрузки для распределения запросов, столкнулась с ошибками при соединении с LDAP-серверами. Причина заключалась в том, что балансировщики нагрузки не закрывали соединения корректно, что вызывало проблемы с обработкой запросов. Решение заключалось в замене оборудования на устройства, корректно выполняющие TCP-команду для закрытия соединений.

### 29.1.6.2. Вертикальное расширение

**Вертикальное расширение** подразумевает улучшение ресурсов одного сервера, а не добавление новых серверов. Это расширение используется для оптимизации производительности на отдельных этапах обработки запросов, например, на серверах, которые выполняют специфические задачи.

#### Пример вертикального расширения:

Предположим, сайт содержит большие видеоклипы и приложение для опросов. Чтение больших видеофайлов и одновременная запись данных в базу данных на одном сервере создают конкуренцию за ресурсы. В этом случае видеоклипы могут быть размещены на отдельном сервере, оптимизированном для обработки больших файлов, а приложение для опросов продолжает работать на первоначальном сервере.

#### Комбинированное расширение:

Горизонтальное и вертикальное расширение могут быть использованы вместе. Например, для сайта с видеоклипами можно добавить дополнительный сервер для обработки видео, а также улучшить производительность сервера для обработки опросов.

### 29.1.6.3. Выбор метода расширения

Выбор метода расширения зависит от характеристик вашего веб-сайта и потребностей в ресурсах. Для этого важно классифицировать компоненты вашего сайта по тому, какие ресурсы они требуют, и определить, какие из них конкурируют за эти ресурсы.

#### Анализ ресурсов:

* **Статические файлы**: Обычно не требуют много ресурсов, но могут быть крупными (например, мультимедиа).
* **CGI-программы**: Могут сильно загружать память или процессор.
* **Базы данных**: Часто требуют значительных системных ресурсов и могут стать "узким местом" в производительности.

#### Пример использования расширения:

* Для сайта с видео и опросами можно переместить видеоклипы на отдельный сервер, а остальные части сайта оставить на исходном сервере.
* Веб-сайт, который также строит графики из логов, может перенести нагрузку от этих графиков на отдельную машину, освобождая основной сервер для других задач.

### Рекомендации по расширению:

* Расширение лучше выполнять поэтапно, начиная с самого перегруженного компонента. Это позволит избежать перегрузки системы и упрощает тестирование.
* Не стоит сразу оптимизировать все компоненты — лучше сосредоточиться на тех, которые создают наибольшую нагрузку.

Таким образом, горизонтальное и вертикальное расширение являются важными инструментами для масштабирования веб-служб. Их можно комбинировать в зависимости от характеристик нагрузки и ресурсов, что позволяет обеспечить стабильность и высокую производительность сайта.

### 29.1.6.4. Проблемы расширения

При расширении подсистем, использующих общие ресурсы, возникают определенные сложности. Например, если веб-сайт поддерживает состояние (например, форма регистрации), то это состояние должно быть доступно на всех системах, которые обрабатывают запросы, или поддерживаться клиентским браузером.

Одной из основных проблем, с которой столкнулись первые системы балансировки нагрузки, было управление состоянием сессий. Современные устройства балансировки нагрузки могут отслеживать виртуальные сеансы, направляя трафик от конкретного клиента к соответствующему серверу. Однако, поскольку запросы могут исходить из-за шлюзов трансляции сетевых адресов (NAT) или брандмауэров, где все запросы выглядят как с одного IP-адреса, это создаёт дополнительные трудности.

Также CGI-программы или скрипты, управляющие информацией, часто используют локальные файлы блокировки для контроля доступа. Если эти программы размещаются на нескольких серверах, лучше изменить их для использования базы данных с блокировками.

Одной из основных проблем с расширением является работа с базами данных. Простое увеличение мощности серверов может быть ограничено, и, несмотря на высокую стоимость, это не решает все задачи. Решение — разделение данных на те, которые только читаются, и те, которые читаются и записываются. Данные только для чтения могут быть продублированы в разных базах данных, а данные для записи — распределены по разным серверам.

Еще одна проблема заключается в том, что страницы могут требовать данных из различных источников. Использование продуктов репликации баз данных (например, Relational Junction) позволяет системным администраторам дублировать таблицы из разных баз данных и объединять их для общего использования.

### Важность расширения

Многие начинают задумываться об оптимизации расширения только тогда, когда становится поздно. Например, в 2000 году сайт избирательной комиссии Флориды не выдержал нагрузки в день выборов, что привело к перегрузке серверов и невозможности доступа к необходимым данным.

Общая схема расширения типичного веб-сайта: начальная загрузка всех компонентов (статические файлы, CGI-программы, базы данных) на одном сервере, затем, с ростом нагрузки, их разделение и перенос на отдельные машины. Статическое содержимое можно легко дублировать, используя сетевое хранилище (например, NFS или SAN). Динамические страницы можно разделить и специализировать, например, страницы с обработкой кредитных карт перенести на отдельный сервер, а страницы каталога — на другой. Базы данных также можно расширять по аналогии.

### 29.1.7.1. Безопасные соединения и сертификаты

Доступ к веб-сайтам чаще всего осуществляется по незашифрованной связи, что подвергает риску конфиденциальность данных пользователей. Для защиты передаваемой информации используется шифрование веб-трафика через протокол SSL (Secure Sockets Layer), который шифрует данные, передаваемые между веб-сервером и клиентом. Это предотвращает перехват данных, особенно когда пользователи подключаются через небезопасные сети, например, в кафе. Адреса сайтов с поддержкой шифрования начинают с https://, что означает использование SSL.

Реализация HTTPS на веб-сервере относительно проста, но управление криптографическими сертификатами может быть сложным. SSL-сертификаты состоят из двух частей: публичной и приватной. Публичная часть доступна всем, кто подключается к серверу, а приватная должна храниться в секрете. Если приватный ключ утечет, злоумышленники могут выдавать себя за сервер. Важно правильно управлять этими ключами, храня их в защищенном месте и используя соответствующие процедуры для их восстановления.

В большинстве случаев частный ключ хранится на сервере в зашифрованном виде и доступ к нему требует ввода пароля. Это создает проблему, так как для перезапуска веб-сервера потребуется человек, который введет пароль. Чтобы решить эту проблему, часто используют метод хранения пароля в зашифрованном виде в защищенной директории, с помощью программы-помощника, которая передает пароль веб-серверу.

Для создания криптографического сертификата используется программа, такая как OpenSSL. Если сертификат самоподписанный, то ему можно доверять только в пределах своей организации. Внешние пользователи могут не доверять такому сертификату. Чтобы удостовериться в подлинности сайта, сертификат должен быть подписан сертифицирующим органом (CA). Клиенты, подключаясь к серверу, могут проверить, был ли сертификат подписан доверенным CA, что подтверждает подлинность сайта.

Процесс шифрования SSL требует интенсивных вычислений, что может замедлить работу сервера. Для решения этой проблемы используются аппаратные ускорители SSL. Веб-сайты используют HTTPS только на критичных страницах, так как шифрование может значительно замедлить обработку запросов.

### 29.1.7.2. Защита приложения веб-сервера

Атаки могут быть направлены непосредственно на веб-сервер, с целью получить несанкционированный доступ к системе. Уязвимости могут быть как в операционной системе, так и в программном обеспечении, расширяющем функциональность сервера, например, в модулях или веб-программировании, выполняемом на сервере. Эти уязвимости нужно регулярно отслеживать и устранять.

Для защиты важно быть в курсе последних уязвимостей и обновлений для веб-сервера и его компонентов. Разработчики веб-серверов и сторонних модулей часто предоставляют списки известных уязвимостей и обновлений через рассылки или форумы. Также рекомендуется использовать системы мониторинга, которые позволяют отслеживать попытки эксплуатации уязвимостей, например, с помощью автоматизированных средств анализа логов.

### 29.1.7.3. Защита содержимого

Веб-сервисы часто становятся целью атак, направленных не на получение доступа к серверу, а на извлечение данных, хранящихся на веб-сайте. Уязвимости безопасности веб-контента могут быть разнообразными, и с каждым годом появляются новые угрозы. Для эффективной защиты важно регулярно отслеживать информацию о последних уязвимостях и проверять безопасность веб-сервера.

Примером таких атак могут быть следующие способы:

* **Обход директорий**: Этот метод направлен на получение доступа к данным, которые не должны быть доступны. Злоумышленник может использовать путь к файлу в виде, например, ../../../некоторый\_файл, чтобы пройти в каталог, который обычно скрыт. Если веб-сервер автоматически индексирует директории, это может привести к тому, что злоумышленник сможет получить доступ к конфиденциальным данным. Современные веб-серверы обычно защищены от таких атак, запрещая доступ к директориям, не указанным явно в конфигурационных файлах. Однако старые или экспериментальные реализации могут быть уязвимы к этому типу атаки.
* **Изменение поля формы**: В некоторых случаях злоумышленники могут изменять поля веб-форм, чтобы получить доступ к данным или изменить функциональность сайта. Например, форма для покупки товара может содержать скрытое поле с ценой товара. Изменив это значение, злоумышленник может установить произвольную цену, что может привести к недопустимым транзакциям, таким как покупка товара по заниженной цене или даже по отрицательной цене, что приведет к компенсации.

Для предотвращения таких атак, лучше использовать уникальные идентификаторы для товаров и привязывать цену к базе данных, а не хранить её в самой форме. В этом случае система будет проверять цену товара в базе данных, а не доверять данным, введенным пользователем.

* **Внедрение SQL (SQL Injection)**: Это более сложный вариант атаки, при котором злоумышленник вставляет свой собственный SQL-запрос в поле ввода формы. Если приложение неправильно обрабатывает данные, введенные пользователем, злоумышленник может выполнить произвольные SQL-запросы, что может привести к удалению или изменению данных в базе, получению привилегированного доступа или даже выполнению команд операционной системы.

Для защиты от SQL-инъекций необходимо правильно фильтровать и экранировать данные, поступающие от пользователей, а также использовать подготовленные запросы (prepared statements), которые минимизируют риск выполнения вредоносных SQL-команд.

Все эти методы требуют внимательности при проектировании веб-приложений и настройки серверов, а также регулярных проверок на наличие уязвимостей, чтобы минимизировать риски потери данных или компрометации системы.

**29.1.7.4. Безопасность приложений**

Для повышения безопасности веб-приложений и снижения риска успешных атак важно соблюдать ряд фундаментальных принципов при разработке кода:

1. **Ограничение потенциального ущерба**: Один из наиболее эффективных методов защиты — ограничить ущерб, который может нанести злоумышленник. Например, хранение содержимого и программ в изолированной внутренней среде и последующая передача их на веб-сервер после проверки может минимизировать последствия. Если веб-сервер изолирован от внутренней сети и не имеет возможности инициировать соединения с другими машинами, злоумышленник не сможет использовать его для атаки на другие системы.
2. **Подтверждение входных данных**: Очень важно тщательно проверять все входные данные, предоставляемые пользователями. Это включает проверку длины данных для предотвращения переполнения буфера и защиты от попыток выполнения произвольных команд через спецсимволы. Лучше всего подходить к фильтрации входных данных с принципом «включения», то есть разрешать только заранее определённые символы.
3. **Автоматизация доступа к данным**: Программы, которые осуществляют доступ к базе данных, должны быть максимально специализированы. Веб-приложение должно открывать базу данных только в режиме чтения, если это необходимо, или использовать предварительно скомпилированные запросы (stored procedures), чтобы избежать выполнения произвольных SQL-команд.
4. **Использование прав и привилегий доступа**: Важно следить за тем, чтобы веб-программы не имели избыточных привилегий. Применение принципа минимальных прав и ограничение доступа через аутентификацию и авторизацию может предотвратить атаки, такие как межсайтовая обратная подделка (XSRF), которая использует несанкционированные привилегии.
5. **Ведение логов**: Логи являются важным инструментом для диагностики и восстановления после вторжений. После атаки злонамеренные лица могут попытаться удалить или изменить логи, чтобы скрыть следы своих действий. Для защиты логов рекомендуется использовать их хранение на отдельных серверах или в нестандартных местах, например, с помощью сетевого ведения логов, что усложняет их подделку и удаление.

**29.1.8. Управление содержимым**

Управление содержимым веб-сайтов должно быть разделено с задачами системного администратора, чтобы избежать создания узких мест в процессе обновления данных. Системные администраторы не должны нести ответственность за регулярное обновление контента, поскольку это отвлекает их от более важных технических задач и приводит к перегрузке.

Для эффективного управления содержимым необходимо:

* Делегировать задачи обновления контента другим сотрудникам, например, контент-менеджерам или веб-мастерам.
* Создать надежный процесс управления содержимым, который не зависит от системных администраторов.
* Обучить пользователей работе с инструментами, позволяющими им самим обновлять контент, например, с помощью средств для форматирования и публикации HTML.

Такой подход помогает снять нагрузку с системных администраторов и повышает гибкость в управлении сайтом, что также способствует повышению безопасности.

**29.1.8.1. Веб-группа**

Для эффективного управления содержимым веб-сайта полезно создавать специализированную веб-группу, которая будет обслуживать как внутренние, так и внешние сайты организации. Это позволяет создать унифицированный процесс обновления контента и вовлечь сотрудников, знакомых с современными методами веб-программирования.

**Преимущества веб-группы:**

* Веб-группа может использовать стандарты и программы для унификации подхода к обновлению контента.
* Сотрудники группы могут углублять свои знания в области веб-программирования, что увеличивает профессионализм.
* Такая группа облегчает делегирование задач и снимает нагрузку с системных администраторов.

Если организация слишком мала для полноценной веб-группы, хорошей альтернативой является создание **веб-комиссии**. В состав комиссии могут войти веб-мастер и представители различных заинтересованных групп, включая системных администраторов. Это помогает:

* Обеспечить ответственность за контент.
* Объединить ресурсы и ускорить обучение.
* Предотвратить перегрузку системных администраторов.

Для системных администраторов, которые занимаются обновлением веб-содержимого, создание веб-комиссии помогает повысить осведомленность о важности их работы и, возможно, получить поддержку для найма отдельного веб-мастера.

**29.1.8.2. Контроль изменений**

Создание веб-комиссии также упрощает процесс контроля изменений на веб-сайте. Веб-комиссия становится естественным владельцем этого процесса, распределяя ответственность за разные типы изменений:

1. **Обновление** – добавление нового контента или замена старой версии документа.
2. **Изменение** – изменение структуры сайта (например, создание новой директории или перенаправление ссылок).
3. **Исправление** – коррекция контента или функциональности сайта, чтобы привести их в соответствие с установленными стандартами.

Процесс исправлений должен включать:

* Оформление заявок на устранение неисправностей.
* Контроль качества, чтобы убедиться, что исправления соответствуют стандартам перед их публикацией.

Процесс изменений должен включать подтверждение от соответствующих участников, например, через письма от членов веб-комиссии, прежде чем изменение будет передано в группу контроля качества.

Такие процессы, аналогичные тем, что используются в инженерных проектах, помогают поддерживать высокое качество и точность в управлении веб-контентом.

29.1.9. Создание типового управляемого веб-сервера

Системных администраторов часто просят создать веб-сервер с нуля, не предоставляя конкретной информации о том, как он будет использоваться. Чтобы точнее определить запрос, полезно задать несколько вопросов, на которые нетехнический пользователь может ответить без необходимости передавать информацию кому-то еще.

Основные вопросы, которые следует задать:

1. Станет ли веб-сервер использоваться только внутренними пользователями или он будет доступен через Интернет?
2. Есть ли у веб-сервера особая задача по размещению конкретной службы или программы? Если да, то какой именно службы или программы?
3. Кто будет пользоваться сервером и какие типичные варианты его использования предполагаются?
4. Каковы требования по безотказной работе? Можно ли отключать веб-сервер для техобслуживания на час в неделю или на шесть часов?
5. Будем ли мы создавать для этого веб-сервера учетные записи или группы?
6. Сколько места для хранения должно быть на сервере?
7. Какой предполагаемый трафик будет поступать на этот сервер и как он будет расти со временем?

29.1.9.1. Любой сайт

Есть несколько ключевых принципов, которые нужно учитывать при планировании веб-сайта, независимо от того, будет ли его использование внутренним или внешним. Один из наиболее важных аспектов — планирование пространства имен URL. Это позволит избежать сложностей при изменении URL в будущем. Например, если веб-сайт меняет название компании, нужно заранее подумать, как изменится URL, чтобы сохранить доступность всех ссылок.

Важно заранее спланировать структуру URL, чтобы избежать конфликтов и обеспечить последовательность. Например, использование идентификационных номеров пользователей вместо имен может привести к конфиденциальности проблемам, если кто-то передаст этот URL другому человеку.

Веб-серверы часто поддерживают функцию перенаправления URL, что позволяет избежать проблем при изменении структуры сайта. Для эффективного управления сайтом можно создать отдельные файлы конфигурации для различных разделов, что поможет избежать ошибок в настройках основного конфигурационного файла.

Если на сайте предполагается размещение контента, важно, чтобы пользователи могли обновлять его. Это можно организовать через доступ к веб-контенту или создание методов безопасного доступа для внешних пользователей. Еще более эффективным решением может быть создание веб-сайта на основе базы данных, что позволит автоматизировать процесс обновления контента.

29.1.9.2. Внутренний сайт

Для внутреннего сайта обычно достаточно простой модели публикации. Создайте корневой каталог документов, к которому можно открыть общий доступ, и предоставьте внутренним группам права на чтение и запись своих поддиректорий. Таким образом, они смогут управлять своим внутренним контентом.

Если внутренним пользователям нужно изменять сам веб-сервер, добавляя модули или директивы конфигурации, которые могут повлиять на другие группы пользователей, рекомендуется использовать отдельный, возможно виртуальный, сервер. Это не нужно делать для каждой группы, но некоторым группам, например, разработчикам, может понадобиться такой подход. Например, разработчики могут требовать изменения веб-сайта с использованием материалов из скриптов установки сторонних программ. В случае факультетов университета, предоставляющих дистанционное обучение, может потребоваться интеграция с сайтом или связь аутентификации с внешними системами.

29.1.9.3. Внешний сайт

Внешние сайты должны настраиваться с учетом безопасности, включая блокировку неиспользуемых портов или использование брандмауэра. Если в вашей организации нет внешнего веб-сайта и сервер, который вы создаете, будет первым, важно уточнить у заказчика согласие на создание сайта и его структуру. Это поможет эффективно распределить время всех участников.

Создание внешнего веб-сайта включает четыре основные компонента: регистрацию домена, хостинг DNS, веб-хостинг и веб-контент.

1. Регистрация домена в глобальном реестре. Эту задачу выполняют провайдеры, или регистраторы.
2. Хостинг DNS — служба, которая обрабатывает DNS-запросы и отправляет ответы. Некоторые регистрационные службы предлагают хостинг DNS как часть своей услуги.
3. Веб-хостинг — это сервер, который вы только что установили, и который соответствует адресу, выданному DNS.
4. Веб-контент — это веб-страницы и скрипты, которые необходимо создать и загрузить на сервер.

29.1.9.4. Процесс создания материалов веб-сайта

Для статичных веб-сайтов, например, для представительства новой компании, рекомендуется использовать процесс ввода в эксплуатацию новых версий сайта. Это может включать три идентичных сервера, каждый для отдельного этапа процесса.

Первый сервер используется для редактирования и загрузки материалов на рабочих станциях. Второй — это сервер контроля качества, на который переносятся материалы для проверки и тестирования. Третий сервер — рабочий сервер, куда перенесены материалы, прошедшие контроль качества.

Если сайт сильно зависит от скриптов или требует строгого контроля контента, можно ввести дополнительный сервер — эталонный. Он функционально идентичен рабочему серверу, но заблокирован от внешнего доступа, либо скрыт за брандмауэром или VPN. Эталонный сервер используется для тестирования приложений или процессов, которые должны беспроблемно взаимодействовать с рабочим сервером. Также он позволяет публиковать контент внутри организации, а затем передавать его группе, отвечающей за размещение на внешнем сайте. Доступ на эталонный сервер ограничен только внутренними пользователями или партнерами.

29.2. Тонкости

До сих пор рассматривались самостоятельные решения. Тонкости же относятся к способам максимального использования внешних служб, чтобы системные администраторы не беспокоились о мелких деталях.

29.2.1. Веб-хостинг третьих сторон

Компании по веб-хостингу предоставляют веб-серверы, которые могут использовать другие пользователи для загрузки и публикации контента. Существует конкуренция за предоставление большего числа функций, большего времени безотказной работы и меньших затрат. Управляемый хостинг — это услуга, когда хостеры предоставляют дополнительные возможности, например, мониторинг.

Крупные компании часто имеют собственную службу управляемого хостинга, чтобы не начинать каждый новый проект с нуля, когда необходимо создать новую веб-службу.

Этот раздел будет полезен системным администраторам, которые обеспечивают работу веб-сайтов или служб хостинга, и касается использования таких внешних служб.

29.2.1.1. Преимущества передачи веб-служб сторонним исполнителям

Интеграция обычно эффективнее, чем создание собственных решений. При использовании услуг хостинга не требуется установка локальных программ — все находится в ведении провайдера. Вместо того чтобы становиться экспертом в области сетевых технологий, установки серверов и инженерных процессов, можно сосредоточиться на управлении веб-службами.

Хостеры часто предоставляют сетевую панель управления, через которую можно управлять размещенной службой. Все данные хранятся на серверах хостера, что может показаться минусом, но на самом деле это приносит преимущества в виде надежности и безопасности, которые гораздо сложнее обеспечить отдельной организацией. Хостеры выигрывают за счет масштабируемости, предоставляя больше резервирования, пропускной способности и ресурсов, чем может позволить себе отдельная организация.

Внешний хостинг веб-приложений или служб помогает организациям более эффективно использовать труд системных администраторов и снижать затраты на оборудование и сетевые ресурсы. Это особенно актуально, когда желаемые услуги требуют значительной индивидуализации или обучения персонала. Услуги управляемого веб-хостинга также могут стать частью плана аварийного восстановления и предоставить дополнительную гибкость при расширении.

Для небольших сайтов использование службы веб-хостинга наиболее экономично, так как хостер часто объединяет десятки таких сайтов на одном сервере. Стоимость может варьироваться от 5 долларов в месяц для сайтов с малым трафиком до нескольких тысяч долларов для сайтов, требующих большого объема пропускной способности.

29.2.1.2. Недостатки передачи веб-служб сторонним исполнителям

Недостатки связаны с беспокойством по поводу данных, трудностями перехода и опасениями, что передача хостинга приведет к передаче системного администрирования.

Что касается данных, то в большинстве случаев их можно экспортировать с размещенного сайта, чтобы сохранить локально. Множество хостеров также предлагают услуги резервного копирования, включая периодическое дублирование данных и отправку копий непосредственно заказчику.

Что касается двух других проблем, многие системные администраторы трудно расстаются с идеей выполнения всех задач самостоятельно, даже если они перегружены. Сохранение ответственности за выполнение всех обязанностей — это одна из лучших форм гарантии занятости, поэтому решения, которые снижают нагрузку, обычно воспринимаются как полезные для работы.

29.2.1.3. Унифицированный вход в систему: управление профилями

Во многих случаях крайне желательно иметь унифицированный, или централизованно организованный вход в системы и приложения организации. Лучше, чтобы все приложения использовали единую систему паролей, чем требовать у сотрудников наличия пароля для каждого приложения. Когда количество паролей становится слишком большим, пользователи начинают записывать их на бумажках или приклеивать к мониторам, что сводит на нет безопасность.

При работе с веб-серверами и приложениями сочетание имени пользователя и пароля, а также дополнительная информация для доступа или индивидуальной настройки обычно называется профилем. Управление профилями на веб-серверах часто представляет собой серьезную проблему. Методы управления профилями для веб-приложений не стандартизированы, и многие современные приложения используют внутреннее управление профилями.

Стандартное веб-приложение либо включает собственный веб-сервер, либо работает под уже существующим. Большинство веб-серверов не предлагают централизованного управления профилями. Вместо этого для каждой директории прописываются правила профилей в конфигурационных файлах веб-сервера. На практике же это часто игнорируется.

Есть несколько стандартных методов, с помощью которых веб-серверы и приложения управляют данными профилей, например, использование файлов .htaccess и .htpasswd в Apache, запросы к LDAP или Active Directory, а также SQL-запросы во внешнюю базу данных. Некоторые приложения поддерживают лишь часть этих методов, а другие используют собственные внутренние системы.

При использовании методов аутентификации, встроенных в программное обеспечение веб-сервера, все вопросы аутентификации обрабатываются до того, как CGI-скрипт получает контроль. Например, в Apache, независимо от того, используется ли аутентификация через локальные текстовые файлы или через более сложные механизмы, такие как модуль аутентификации LDAP, запрос на ввод имени пользователя и пароля обрабатывается на уровне веб-сервера. CGI-скрипт запускается только после успешного входа в систему, и ему передается имя пользователя через переменную окружения.

29.2.2. Гибридные приложения

Один из побочных эффектов стандартных форматов обмена данными между веб-приложениями — это явление, называемое гибридными приложениями. Такие приложения могут привести к серьезным проблемам с масштабированием.

Гибридное приложение — это веб-сайт, который использует данные и API других веб-сайтов для создания нового приложения. Гибридные приложения извлекают данные из структурированных выходных данных веб-служб и вставляют их в новое приложение. Это сочетание может быть очень полезным и разносторонним. Разработчики создают сложные XML-схемы для данных своих приложений.

Примером гибридного приложения является HousingMaps, который использует данные Google Maps и информацию о недвижимости с сайта Craigslist.

Гибридное приложение состоит из двух частей: первая — это сам компонент гибридного приложения, написанный разработчиком для интеграции различных служб, а вторая — используемые службы. Расширение первой части может быть относительно простым, но проблема возникает, когда приложение становится популярным и увеличивается нагрузка на инфраструктуру веб-сервера.

Основная проблема заключается в службах, на которые полагается гибрид. Эти службы выполняют тяжелую работу, и внезапная популярность гибридного приложения может вызвать резкое увеличение количества запросов. Хороший API должен ограничивать и управлять количеством запросов. Например, большинство API требуют от пользователя регистрации для получения уникального идентификационного ключа, который передается в каждом запросе. Этот ключ обычно позволяет делать определенное количество запросов в секунду и ограничивает общее количество запросов в течение 24 часов.

29.3. Заключение

Сеть становится все более универсальной системой для предоставления услуг в организациях, а веб-браузер становится общим интерфейсом для взаимодействия с множеством приложений. Веб-службы должны проектироваться с учетом возможности их расширения, как горизонтально (добавление копий), так и вертикально (распределение нагрузки между различными уровнями). Хотя создание простого веб-сайта для различных приложений относительно легко, гораздо сложнее обеспечить, чтобы системные администраторы не были перегружены обслуживанием содержимого.

Одним из решений может стать создание комиссии из заинтересованных лиц, включая выделенную группу веб-мастеров, что позволит решать вопросы расширения и обеспечит гибкость в использовании веб-служб. Кроме того, для некоторых простых или ресурсоемких служб можно рассмотреть вариант передачи их сторонним исполнителям, что позволит системным администраторам сосредоточиться на более приоритетных задачах.

Глава 30. Организационная структура

Организационная структура группы системного администрирования и ее интеграция в общую структуру компании играют решающую роль в успехе или неудаче работы данной группы. В этой главе рассматриваются ключевые вопросы, которые должны быть учтены при создании эффективной группы системного администрирования, а также представлены результаты наблюдений в разных компаниях. Заключение главы включает примеры различных организационных структур.

Общение — это область, которая напрямую зависит от организационной структуры как группы системного администрирования, так и всей компании. Структура компании определяет ключевые каналы общения как между системными администраторами, так и между ними и пользователями. Хорошо организованные каналы общения являются основой для успешной работы группы системного администрирования, поскольку это позволяет системным администраторам работать эффективно, строить надежную инфраструктуру и предоставлять качественную поддержку пользователям.

Одной из важнейших задач является создание структуры, которая будет способствовать сотрудничеству, а не возникновению конфликтов. Слабые каналы общения внутри группы или между системными администраторами и пользователями могут привести к противостоянию и ухудшению работы. Даже если организационная структура предполагает конкуренцию или противостояние, грамотное управление коммуникациями может избежать возникновения конфликта «они против нас».

### 30.1. Основы

Эффективная организация системного администрирования требует внимательного подхода к размеру и финансированию отдела. Группа должна предоставлять высокий уровень обслуживания без излишних затрат. Определение правильного размера отдела и бюджета, поддержание баланса между качеством и стоимостью обслуживания — это непростая задача. Важным фактором также является роль руководства в организации работы группы.

Идеальная группа системных администраторов способна обеспечить нужный уровень обслуживания при минимальных затратах. Ключевым является не просто увеличение численности команды, а привлечение людей с нужными навыками. Хорошая группа должна быть технически грамотной и уметь эффективно общаться внутри коллектива и с остальными сотрудниками компании.

Для небольших групп системного администрирования важна многозадачность: администраторы должны иметь широкий спектр знаний и навыков. Рассматривается, какие функции лучше централизовать, а какие — распределить между маленькими группами. Описание централизованных и децентрализованных моделей работы поможет выбрать оптимальный подход для организации работы системных администраторов.

**30.1.1. Определение размеров**

Правильный размер группы системного администрирования — ключевой фактор для эффективной работы. Если группа слишком маленькая, она может не справиться с нагрузкой, что приведет к проблемам с инфраструктурой и поддержкой пользователей. Если группа слишком большая, компания будет нести лишние расходы, а коммуникация между администратором станет сложной. На практике чаще возникает ситуация, когда группы недоукомплектованы. Слишком большое количество сотрудников часто связано с нехваткой необходимых навыков в организации.

Если группа испытывает трудности с поддержкой пользователей, просто увеличение численности может не решить проблему. Вместо этого нужно задуматься о приобретении недостающих навыков через тренинги или консультации.

При определении размера группы важно учитывать:

* количество и разнообразие пользователей и оборудования,
* сложность работы,
* уровень обслуживания, который требуется различным группам,
* критичность стабильности работы компьютерных систем.

Для оценки рабочих нагрузок можно опросить системных администраторов, чтобы понять, сколько времени они тратят на различные задачи. В идеале система уведомлений о неисправностях должна быстро предоставлять эту информацию, а в случае ее отсутствия можно попросить администраторов заполнить форму. Это поможет определить темпы роста группы и позволит оптимизировать расходы на поддержку.

**30.1.2. Модели финансирования**

Как и кто финансирует отдел системного администрирования, играет решающую роль в его успехе или неудаче. Системное администрирование часто воспринимается как расходный, а не доходный отдел, что может привести к его недофинансированию. Однако, когда инфраструктура и поддержка начинают ухудшаться, это ведет к большим потерям, поскольку сотрудники тратят много времени на решение проблем с компьютерами и сетями.

Чтобы избежать такого состояния, важно выбрать правильную модель финансирования и ответить на вопросы:

* Кто платит?
* Как это соизмеряется с потребностями?
* Что пользователи получают за свои деньги?

Существует два основных типа моделей:

* **Централизованная модель**, где отдел финансируется компанией в целом, и его работа контролируется централизованно. Это позволяет контролировать расходы и использовать экономию на масштабе.
* **Децентрализованная модель**, где отдельные подразделения бизнеса финансируют системное администрирование. Это дает подразделениям больше контроля и позволяет лучше адаптировать обслуживание под их нужды.

Перемещение между этими моделями может быть сложным и болезненным для всех сторон. Когда компания решает перейти от одной модели к другой, важно проводить открытые встречи с системными администраторами, чтобы объяснить изменения и выслушать их мнение. Вовлечение всех сторон в этот процесс помогает избежать проблем и сделать переход более плавным.

Для достижения успеха важно выбрать правильную модель с самого начала или проводить регулярные улучшения, вместо того чтобы надеяться, что кардинальные изменения решат все проблемы.

### 30.1.3. Влияние цепи управления

Цепь управления в организации системного администрирования имеет большое значение для её работы. В быстро развивающихся компаниях информационные технологии могут находиться в ведении главного технического директора, который также ответственен за инженерные и исследовательские отделы. В других компаниях системное администрирование может подчиняться хозяйственному отделу и отчитываться перед генеральным или финансовым директором.

Если информационный директор отчитывается перед техническим директором, компания рассматривает IT как инвестицию для получения прибыли, а если перед финансовым директором — как расход, который нужно минимизировать.

**Технический директор и инженерное управление**: Когда системное администрирование подчиняется техническому директору, это может приводить к тесным отношениям между системными администраторами и инженерным отделом, что повышает финансирование и приоритетность IT-проектов в этой области. Однако другие части компании могут страдать из-за того, что приоритеты будут отдаваться проектам инженерной группы, а не общей инфраструктуре.

**Исполнительный или финансовый директор**: Если системное администрирование подчиняется исполнительному или финансовому директору, это может привести к более сбалансированному и беспристрастному распределению ресурсов, так как они имеют более широкий взгляд на компанию в целом. Однако они могут не полностью понимать потребности IT-отдела, что затрудняет точную оценку необходимых инвестиций.

Каждая модель имеет свои плюсы и минусы, и выбор подходящей зависит от специфики компании и ее потребностей. Важно, чтобы руководство осознавало, как структура отчетности влияет на работу группы системного администрирования и использовало сильные стороны существующей структуры, минимизируя её слабости.

### 30.1.4. Подбор навыков

Создание эффективной команды системных администраторов требует собирательства самодостаточной группы с разнообразными наборами навыков и ролями. Обязанности системных администраторов можно разделить на четыре основные группы:

1. **Техническое обслуживание и поддержка клиентов**: включает службу поддержки пользователей, обслуживание серверов, поддержку второго уровня и специализированную поддержку для отдельных групп пользователей.
2. **Внедрение новых служб**: эти специалисты фокусируются на новых проектах и не должны отвлекаться на срочные запросы от пользователей.
3. **Проектирование архитектуры новых служб**: это системные архитекторы, которые разрабатывают новые технологии, проектируют и создают прототипы новых служб для пользователей.
4. **Широкий профиль специалистов (специалисты по интеграции)**: они обладают глубокими знаниями о взаимодействии различных систем и могут решать сложные проблемы, требующие междисциплинарного подхода.

В малых компаниях эти роли часто пересекаются, и один человек может исполнять несколько обязанностей. В крупных организациях эти должности разделяются, и для системных администраторов создаются возможности для профессионального роста, например, через ротацию в разных областях.

Работа в службе поддержки полезна для младших системных администраторов, так как помогает понять распространенные проблемы и обучать менее опытных коллег. Важно, чтобы сотрудники также имели возможность работать с конструкторами и проектировщиками, что помогает им лучше понимать потребности пользователей и улучшать взаимодействие между различными частями команды.

Через такой подход можно повысить общую эффективность работы команды и обеспечить лучший сервис для пользователей компании.

### 30.1.5. Инфраструктурные группы

С ростом компании появляется необходимость в специально выделенной команде для поддержки инфраструктуры. Эта группа отвечает за централизованные службы, такие как аутентификация, печать, электронная почта, DNS, календарные службы, сети, удаленный доступ, директории и безопасность. Также она часто управляет автоматизированными системами для системных администраторов, включая автоматическую загрузку, конфигурацию и установку патчей на новые машины.

Команда инфраструктуры должна быть единой, даже если она распределена по разным локациям. Инфраструктура компании должна быть согласованной, а если различные группы управляют разными ее частями, то могут возникать проблемы с совместимостью протоколов и интерфейсов, что приведет к сбоям в системе. Чтобы избежать этих проблем, рекомендуется создавать централизованную группу инфраструктуры, которая будет отвечать за рост и поддержание всех систем. В случае, когда другие подразделения нуждаются в собственных инфраструктурных решениях, их администраторы должны отчитываться перед центральным руководством, а не местным.

Примером может служить организация единого подхода к электронной почте в компании. Согласованное использование форматов и серверов почты помогает в создании профессионального образа компании и улучшает безопасность. Это также упрощает администрирование почтовых систем и улучшает общую безопасность через централизованную фильтрацию и защиту.

Кроме того, безопасность компании зависит от сильных мест в инфраструктуре. Если разные группы используют разные стандарты и подходы, это может создать уязвимости, которые могут быть использованы злоумышленниками.

### 30.1.6. Поддержка пользователей

Поддержка пользователей — это важнейшая часть работы системного администрирования. Она направлена на обеспечение эффективной работы пользователей, что включает удовлетворение всех их компьютерных потребностей. Поддержка пользователей работает наиболее эффективно в распределенной модели, в отличие от инфраструктуры, где предпочтительнее централизованный подход.

Когда для поддержки пользователей выделяется специальный персонал, это позволяет лучше соотнести время отклика с ожиданиями пользователей. Ключевым моментом здесь является установление персональных отношений между пользователями и их системными администраторами, что помогает повысить доверие и эффективность работы.

В компании могут быть использованы разные модели поддержки. В децентрализованной модели каждая группа или подразделение имеет своих системных администраторов, и связь с пользователями обычно более тесная. Однако в таких случаях часто страдает координация между системными администраторами разных подразделений, что может приводить к проблемам с автоматизацией задач, а также к дублированию работы.

Централизованная модель, в свою очередь, приводит к более сильной инфраструктуре, но может ослабить связь с пользователями, поскольку системные администраторы не всегда знают особенности работы пользователей в разных подразделениях. В такой модели пользователи могут столкнуться с тем, что каждый раз общаются с разными специалистами, и отсутствие персонализированного подхода может снижать удовлетворенность.

Для максимальной эффективности в централизованной модели желательно, чтобы в группу поддержки входило не более пяти системных администраторов, которые будут тесно работать с определенным кругом пользователей. Это помогает создать хорошие отношения с пользователями и повысить эффективность работы.

Для крупных организаций может быть использована комбинированная модель, когда в каждом подразделении есть несколько системных администраторов, которые подчиняются центральному руководству. Однако такая модель может иметь и недостатки обеих моделей — централизованной и децентрализованной.

В любом случае важно учитывать недостатки каждой модели и стараться минимизировать их влияние.

### 30.1.7. Служба поддержки

Первая линия службы поддержки пользователей (первичный уровень) эффективнее всего работает при централизованной организации. Эта служба принимает первичные запросы пользователей, решает базовые вопросы и перенаправляет их в соответствующие отделы, если требуется более глубокое вмешательство. Пользователи хотят иметь единый контакт для обращения в службу поддержки — телефон, адрес электронной почты или веб-страницу, чтобы не тратить время на выяснение, какое подразделение должно решить их проблему.

Для крупных компаний необходимы хорошо скоординированные региональные службы поддержки, чтобы каждое значительное подразделение имело собственную службу, но при этом все они должны быть связаны. В таких компаниях важно найти баланс между эффективной центральной службой поддержки, которая может принимать все обращения, и маленькими распределёнными службами, которые обеспечивают более персонализированное обслуживание, но не способны обработать все запросы.

### 30.1.8. Аутсорсинг

Передача функций системного администрирования сторонним компаниям может быть подходящим решением для организаций, не для которых IT не является ключевой частью бизнеса. В таком случае, компания может подписать контракт с аутсорсером и не беспокоиться о найме, зарплатах или удержании сотрудников. Этот подход хорош для малых компаний, у которых есть базовые потребности в компьютерах, так как они не обязаны искать специалистов для оценки знаний и навыков системных администраторов.

Однако крупным компаниям, для которых IT и системное администрирование — это важные части бизнеса, аутсорсинг может не подходить. Например, для компаний, занимающихся электронной коммерцией, IT является основой их бизнеса, и поддержка сайтов и систем высокой доступности критически важна для стабильной работы. В таких ситуациях аутсорсинг может привести к проблемам с контролем качества и безопасности, особенно если сторонний поставщик не соблюдает высокие стандарты.

Безопасность также представляет собой ключевую проблему при аутсорсинге. Если компания обрабатывает конфиденциальную информацию, важно обеспечить строгие соглашения о конфиденциальности с поставщиками услуг. Внутренние службы безопасности заинтересованы в успехах компании, а поставщики услуг могут быть ориентированы исключительно на выполнение условий контракта, что не всегда соответствует интересам бизнеса.

Для крупных организаций с интернет-сервисами аутсорсинг может быть выгодным для хранения оборудования в аппаратных центрах провайдеров, которые обеспечивают высокую надежность интернет-соединений и энергоснабжения. Однако при этом важно чётко прописывать уровни обслуживания в контракте, включая гарантированную пропускную способность, время отклика, время непрерывной работы и физическую безопасность.

Аутсорсинг — это сложный выбор, и для разных типов организаций у него есть свои плюсы и минусы. Вопросы безопасности и контроля качества должны быть особенно тщательно проработаны.

### 30.2. Тонкости

Организация системного администрирования требует внимательного подхода к использованию консультантов и подрядчиков, что может помочь в расширении команды и строительстве новых служб и инфраструктуры. Однако неправильное использование таких специалистов может привести к демотивации и отчуждению команды.

**Консультанты и подрядчики:**

* **Консультанты** — это эксперты, привлекаемые для выполнения специфических задач, таких как проектирование и внедрение новых сервисов или инфраструктуры, где необходимы глубокие знания. Они часто используются для оказания помощи в специфических проектах, где внутренние системные администраторы могут не обладать нужным опытом.
* **Подрядчики** — это специалисты, которые выполняют задачи, для которых у компании уже есть достаточные знания и навыки, таким образом, их работа фактически дублирует усилия внутренних сотрудников.

Хорошие отношения с консультантами могут приносить большую пользу, если они работают вместе с внутренними системными администраторами, делясь своими знаниями и обучая сотрудников. Однако, важно избежать ситуации, когда консультанты работают над новыми интересными проектами, оставляя внутреннюю команду заниматься только рутинной поддержкой, что может демотивировать сотрудников.

Если задача требует создания новых решений, но внутренние администраторы не могут выделить время на их реализацию, можно привлечь подрядчиков для выполнения повседневной работы. Это даст возможность системным администраторам заниматься более сложными и интересными проектами. Чередование участников в таких проектах также способствует росту и укреплению команды.

### 30.3. Примеры организационных структур

Организация системного администрирования будет сильно зависеть от размера компании и ее специфики, например, компании электронной коммерции или образовательные учреждения. Рассмотрим, как могут быть организованы такие службы в разных типах организаций.

#### 30.3.1. Малая компания

В малой компании (20–200 сотрудников) будет всего 1-2 системных администратора, которые будут отвечать за все базовые потребности компании, включая поддержку пользователей и обслуживание инфраструктуры. Не будет формальной службы поддержки, и запросы будут поступать через электронные каналы. Системные администраторы также будут участвовать в проектировании и реализации новых служб, если это необходимо.

Когда компания растет до более 200 сотрудников и превращается в компанию среднего размера, начнется формирование официальной службы поддержки, а задачи системных администраторов станут более специализированными.

#### 30.3.2. Компания среднего размера

В компании среднего размера (200–1000 сотрудников) начинается специализация работы системных администраторов. Это позволяет создать более структурированную и эффективную службу поддержки, которая будет решать более сложные задачи. В этой стадии могут появиться отдельные специалисты по операционным системам, сетям и безопасности, и постепенно группа станет более специализированной.

При этом важно, чтобы архитекторы и дизайнеры систем могли работать над многозадачными проблемами, которые включают разные технологии. В этой компании может существовать гибридная модель, когда сотрудники службы поддержки также вовлечены в проектирование и развитие новых систем, в том числе временно переключаясь с задач поддержки пользователей на участие в разработке.

**30.3.3. Крупная компания**

В крупной компании организационная структура системного администрирования будет гораздо более специализированной и разделенной на несколько ключевых групп. Это может включать:

* **Служба поддержки** с четко определенными уровнями, например, группа поддержки второго уровня для решения более сложных проблем.
* **Малые группы** по работе с пользователями, которые могут быть ориентированы на конкретные отделы или бизнес-подразделения.
* **Централизованные группы** для обслуживания инфраструктуры и обеспечения безопасности, обеспечивающие стабильность и защиту данных компании.
* **Архитекторы и разработчики** на каждое технологическое направление, работающие над проектированием и внедрением новых решений.

Кроме того, могут существовать **региональные организации системного администрирования**, или отдельные группы, ориентированные на поддержку филиалов или крупных подразделений. Для координации работы этих групп потребуется наличие официальных каналов связи и четко прописанных областей ответственности, которые должны соответствовать общей политике компании.

**30.3.4. Компания электронной коммерции**

Организация системного администрирования в компании электронной коммерции имеет свои особенности, так как существует необходимость в обслуживании двух отдельных групп: **внешних систем** (связанных с интернет-присутствием) и **внутренних корпоративных систем**. Эти группы должны обслуживать разные запросы с различными уровнями доступности и критичности:

* **Группа для интернет-присутствия** обслуживает все внешние запросы, которые напрямую влияют на прибыль компании. Проблемы с внешними системами всегда имеют приоритет.
* **Группа для корпоративных систем** ориентирована на внутренние задачи и поддерживает корпоративную инфраструктуру.

Разделение этих двух направлений помогает избежать конфликтов интересов и повышает эффективность работы. В таких компаниях интернет-служба часто не включает мелкие группы поддержки для каждого подразделения, а состоит из системных администраторов и архитекторов, которые поддерживают и развивают веб-инфраструктуру.

**30.3.5. Университеты и некоммерческие организации**

Университеты и некоммерческие организации часто сталкиваются с ограничениями бюджета, что требует оптимизации использования имеющихся ресурсов. В университетах особенно важна **централизация системного администрирования**, чтобы все подразделения и группы работали в рамках общей системы.

* **В университетах** могут возникать «сферы интересов», где каждое исследовательское подразделение или группа пытается получить финансирование на собственные нужды. Это может приводить к дублированию усилий и недостаточной координации.
* В **некоммерческих организациях** также важна экономия, и управление системным администрированием должно быть максимально централизованным для снижения затрат и улучшения взаимодействия между департаментами.

Руководители таких организаций должны обладать высокими управленческими навыками и быть готовы обеспечить эффективную работу центральной группы системного администрирования.

**30.4. Заключение**

**Оптимизация расходов** на системное администрирование и правильное **распределение ресурсов** играют ключевую роль в эффективной организации. Важно правильно сбалансировать бюджет, чтобы поддерживать нужный уровень поддержки и технологической инфраструктуры, не нарушая других внутренних процессов компании.

**Централизация и децентрализация** — еще один важный момент в организации. Централизованное администрирование инфраструктуры обычно обеспечивает стабильность и надежность, в то время как поддержка пользователей может выигрывать от более гибкой, децентрализованной модели. Важно обеспечить высокий уровень **взаимодействия между системными администраторами** и своевременно решать возникающие конфликты на низших уровнях, чтобы не перегружать руководство компании.