**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Разработка, сопровождение и обеспечение безопасности информационных систем |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Эссе**

**Тема:** Сервер (Server).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **по дисциплине** |  | Администрирование информационных систем | | |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | **Каримов Мухаммад Я** |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | **Вбио-302-рсоб** |  |
|  |  |  |  |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев Иван Валерьевич |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024 г**

**Что такое сервер?**

Что многие представляют себе когда слышат слово "**сервер**"? Скорее всего, какой то мощный и большой компьютер, который стоит специальном помещении, дата-центре, среди множества таких же компьютеров. И, конечно, все эти компьютеры выполняют задачи чрезвычайной важности, буквально "управляют миром". Однако, это не совсем так. Реальность более многообразна и интересна.

*Я обратил внимание, что на канале очень мало статей ориентированных на далеких от техники людей (просто, они занимаются другими важными делами) или школьников, тем более, не старшеклассников. Но ведь как раз школьники активно изучают окружающий мир. И всем может быть любопытно, а как все устроено...*

Сегодняшняя статья как раз рассчитана, большей частью, на школьников и тех читателей, которые в обычной жизни, далеки от информационных технологий или, сокращенно, ИТ.

**Немного этимологии**

В латинском языке есть слово **servitium** которое переводится как **услуга**. Как известно, на латыни разговаривали в Древнем Риме. А Римская Империя оказала очень большое влияние не только на Европейскую цивилизацию, но и на весь мир.

Не удивительно, что во многих языках есть слова, так или иначе связанные с латинским servitium. В английском языке это serve и service, в испанском servicio, в итальянском servizio, во французском service и servir. Думаю вы уже заметили, что слово сервер, как и англоязычное server, родственно этим словам. И это вовсе не случайно.

И слово **server** можно перевести как **обслуживающий**. Или оказывающий услугу. При этом не стоит воспринимать "оказывающего услугу" как простую прислугу! Это совершенно не связанные понятия.

Получается, что **сервер не командует**, не управляет миром, а лишь **обслуживает**, **оказывает услуги**. Означает ли это, что мы свергаем серверы с того пьедестала, на котором они находятся по представлениям не такого и малого числа людей? Вовсе нет! Кроме того, это лишь слово, термин. А что скрывается за ним?

**Наша обычная жизнь**

Итак, само слово, понятие, уходит корнями в древность, когда никаких компьютеров не было и в помине. В глубокую древность мы не пойдем, даже в Древний Рим заглядывать не будем. Но примеры из обычной жизни, которые имеют к вопросу непосредственное отношение, рассмотреть необходимо. Для лучшего понимания.

Предположим, что человек проголодался. Как он может поступить? По большому счету, вариантов не так и много. Он может приготовить еду сам или обратиться к тому, кто приготовит еду для него.

**Делаем все сами**

В самом простом случае, можно приготовить еду самостоятельно. например, может приготовить еду сам. Схематически это можно изобразить примерно так

Человек обслуживает себя сам. Иллюстрация моя

В этом случае он ни к кому не обращается и обслуживает себя сам. С точки зрения темы сегодняшней статьи нам этот случай не интересен.

А вот вариант "обратиться к кому то другому" не только более интересен для нас, но и позволяет выделить два отдельных случая.

**Обращаемся к повару**

Если человек не умеет, или не хочет, готовить, он может обратиться к кому то другому, кто умеет готовить. Например, к уличному торговцу готовой едой или в небольшое кафе, где есть только повар. Для определенности будем считать, что он обратился к повару. Схематически это может выглядеть примерно так

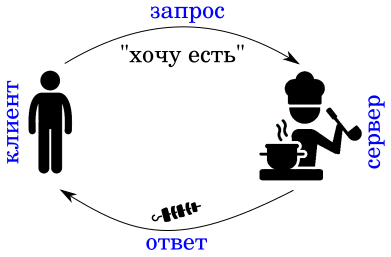
Человек обращается к повару для приготовления еды. Повар обслуживает человека, оказывает ему услугу "приготовление еды". Иллюстрация моя

На первый взгляд, изменилось не многое. Велика ли разница, что готовит еду? Не очень велика, с точки зрения обеда, но при этом у нас появляется весьма важная особенность. Повар оказывает человеку услугу, обслуживает его. Процесс обслуживания я специально выделил на иллюстрации.

Посетителя кафе обычно называют **клиентом**. А вот повара, который в данном случае обслуживает клиента, можно назвать **сервером**! Конечно, мы обычно не называем повара сервером. Но, по сути, он выполняет именно эту функцию. И отношения повара и человека, в данном случае, это именно отношение клиент-сервер. Непривычно? Но это именно так.

Почему человек стал клиентом, а повар сервером? Потому что человек запрашивает (получает) обслуживание. А повар это обслуживание выполняет, предоставляет услугу.

Давайте рассмотрим процесс обслуживания чуть подробнее



Взаимодействие между клиентом и сервером осуществляется через ответы на запросы. Иллюстрация моя

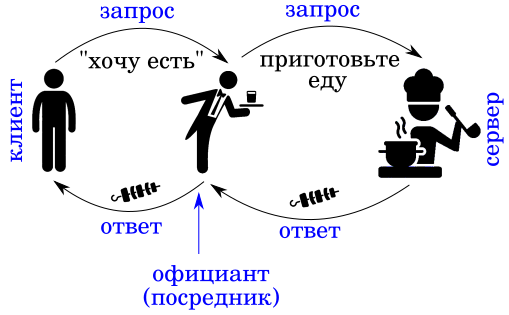
Человек (клиент) обращается к повару (сервер) с запросом "принесите поесть". Повар (сервер) обрабатывает запрос, в данном случае, готовит еду. И передает ее (ответ) человеку (клиент). Запрос - ответ. Это основа взаимодействия.

Но ведь возможна и ситуация, когда инициатива исходит от повара. Например, он зазывает посетителей словами "Самый лучший шашлык! Зайди и убедись сам!". На первый взгляд, здесь запрос и ответ поменялись местами. Но это не так. Повар **анонсирует** свою услугу, сообщает о своих возможностях. И человек по прежнему обращается с запросом услуги, даже если этот запрос выражается словами "хорошо, я попробую".

И этот момент для окажется очень важным в дальнейшем.

**Идем в ресторан**

Казалось бы, какая разница? Но она есть. В ресторане человек не обращается непосредственно к повару, он пользуется услугами посредника - официанта



Между человеком-клиентом и поваром-сервером появился официант-посредник. Иллюстрация моя

И эта ситуация действительно интересна и важна для нас. Смотрите, с точки зрения клиента сервером выглядит именно официант. Ведь ему, а не непосредственно повару, он передает запрос. И от него же получает ответ в виде обеда. Но официант (посредник) не выполняет запрос сам, он передает его серверу или другому посреднику, цепочка может быть длинной.

При этом посредник имеет право и изменить первоначальный запрос клиента. В нашем примере он заменил "хочу есть" на "приготовьте еду". Но изменения могут быть и более значительными. Например, официант может передать повару и подробный список блюд, которые нужно приготовить, а не просто "приготовьте еду". И может добавить уточнения, например, "клиент очень вредный", это уже от себя.

Так же, посредник может собрать несколько ответов сервера (повара), например, несколько блюд, в единый ответ, например, разместить все тарелки на одном подносе, и уже его передать клиенту.

Кроме того, посредник может разбить запрос клиента на несколько отдельных запросов, которые передаст разным серверам (поварам). И потом соберет ответ для клиента из результатов работы разных серверов.

На самом деле, посредники могут быть разными. И для их обозначения могут использоваться разные термины. Например, брокер, собственно посредник. Или прокси, промежуточный. Это не является темой сегодняшней статьи. Поэтому будем и дальше использовать термин посредник.

Возникает интересный вопрос, можем ли посредника называть сервером? Да, конечно можем. Но все зависит от того, о каком именно обслуживании (услуге) идет речь. В данном случае, когда речь идет именно о приготовлении еды, сервер это повар, а официант именно посредник. Но если требуется услуга не "приготовить еду", а "принести еду", то ее предоставлять будет именно официант, или курьер. И именно он будет являться сервером.

**Сервер это не слуга, а клиент не господин**

Очень важно отметить, что **отношение клиент-сервер касается лишь отношения действий и никак не устанавливает отношения подчиненности**! Если вернуться к нашим примерам, и человек, и повар, и официант, могут иметь равные статусы Боле того, статус, например, повара может быть даже выше статуса человека пришедшего пообедать.

Клиент отправляет серверу запрос, но это обращение, можно сказать, просьба, а отнюдь не команда. Более того, сервер, в нашем случае повар, может установить свои требования как к запросам, так и к клиентам. Например, он может не пускать на обед людей, которые неряшливо одеты. И может принимать запросу лишь на некоторых языках, так как другие просто не знает. Или принимать запросы только в письменном виде, например, если у него проблема со слухом.

Другими словами, **отношения клиент-сервер это отношения заказчик-исполнитель**, а не начальник-подчиненный. И, тем более, не господин-слуга.

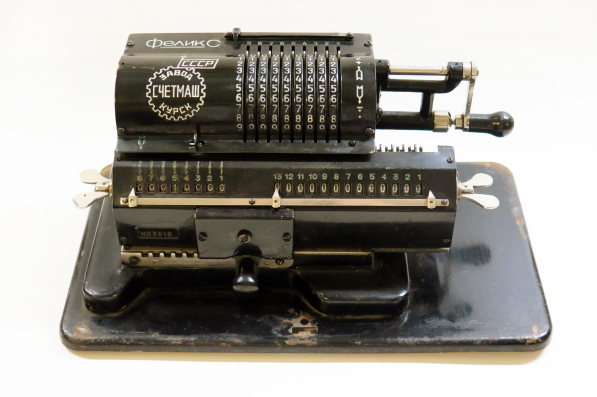
Однако, отношения клиент-сервер не запрещают и установления каких либо дополнительных отношений. Например, слуга может являться сервером. Но сервер совсем не обязательно слуга. А вассал может обратиться к сюзерену с запросом о разрешении строительства дома.

**В мире компьютеров**

Примеры из нашей обыденной жизни позволяют лучше понять суть и смысл отношений клиент-сервер. Однако, нас сегодня больше интересует все таки мир компьютеров.

**Когда компьютеры были большими... И одинокими**

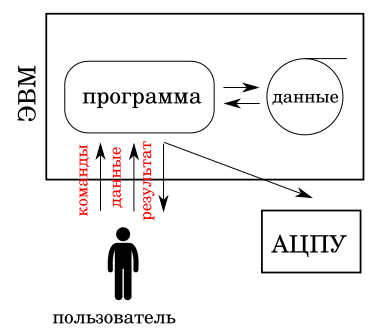
То, что мы сегодня называем обобщенным термином компьютер, начиналось как чисто вычислительное приспособление. Причем появилось оно до начала эры электроники. Вспомните логарифмическую линейку или механический арифмометр, прозванный в народе Железным Феликсом



Арифмометр Железный Феликс это тоже компьютер, только механический. Изображение с сайта belkino.livejournal.com

Появление электроники всего лишь сделало процесс вычислений не механическим, а электронным. Появились Электронные Вычислительные Машины, сокращенно ЭВМ. Долгое время каждая машина была независима от других. Она могла решать задачи одного пользователя в каждый момент времени, или сразу нескольких пользователей.

Можно условно изобразить процесс взаимодействия человека и машины при решении задачи примерно так



Взаимодействие человека и машины на ранних этапах развития вычислительной техники. Иллюстрация моя

АЦПУ это Алфавитно-Цифровое Печатающее Устройство. Достаточно типичное устройство вывода результатов работы больших ЭВМ. Сегодня мы больше привыкли называть его принтером.

Обратите внимание, что на самом деле пользователь взаимодействует не с машиной, не с компьютером, а с выполняемой машиной программой. Не смотря на то, что внешне это выглядит именно как взаимодействие с машиной. Немного позже нам понадобится этот небольшой нюанс.

Человек отправляет машине команды и предоставляет данные. В ответ получает результаты вычислений, которые могут дополнительно выводиться на печать. При этом машина и сама хранит некоторые данные используемые для решения задачи. Например, различные справочники и таблицы.

По сути ,такое использование ЭВМ, компьютера, ничем не отличается от самостоятельного приготовления еды, как мы ранее уже рассматривали. Однако, здесь может быть несколько важных нюансов, которые нам чуть позже понадобятся.

**Компьютеры объединяются в сети**

Компьютерные сети, сначала локальные, а потом и глобальные, появились из-за возникновения необходимости во взаимодействии отдельных компьютеров. Причем не только для передачи данных, но и их совместного использования.

И именно в сетях стали появляться компьютеры-серверы. Но какие услуги, какое обслуживание, может предоставлять компьютер? Он ведь не приготовит обед. Но, конечно, это и не требуется. Зато он может, например, предоставлять услуги хранения данных. Или услуги высококачественной печати документов.

Хранение больших объемов информации требует использования дорогостоящих накопителей (диски, ленты). Сегодня цены не кажутся очень высокими, но когда то такие накопители стоили целое состояние. Информация чаще всего хранится в виде файлов, что и дало название таким серверам - файловый сервер.

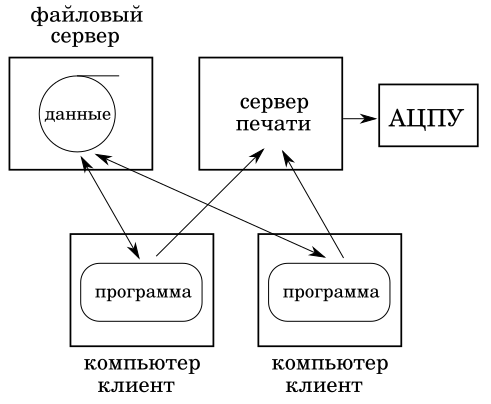
Файловый сервер предоставляет не только услугу хранения данных, что можно было бы считать некоторым аналогом камеры хранения. Такой сервер предоставляет услугу **совместного использования данных**. Ведь с запросом на доступ к данным могут обратиться различные компьютеры-клиенты, причем даже одновременно.

Что бы избежать конфликтов при совместной работе с данными файловый сервер устанавливает определенные правила доступа. Вспомните, мы уже о таком праве сервера на примере повара. И следит, что бы эти правила соблюдались. Это очень просто, если правила не соблюдаются, то запрос отвергается, а не выполняется.

Точно так же, высококачественный высокопроизводительный принтер будет дорогим. А значит, его не имеет смысла устанавливать у каждого компьютера. Мы можем подключить его к серверу, который будет оказывать в сети услугу печати. И называться он будет, естественно, сервером печати (принт-сервером).

Сервер печати не просто предоставляет услугу печати, он предоставляет услугу **совместной печати**. На самом деле, к такому серверу может быть подключен и самый обычный принтер, просто он будет один на несколько компьютеров-клиентов. И он, так же, как и файловый сервер, устанавливает свои правила использования функции печати.

Файловые серверы и серверы печати были, да и остаются, популярными и наглядными иллюстрациями превращения одних компьютеров в клиенты, а других в серверы.



Выделение отдельных компьютеров для выполнения функций файловых серверов и серверов печати. Иллюстрация моя

Давайте еще раз остановимся на **очень важной** подробности. Файловый сервер и сервер печати не просто предоставляют в общий доступ данные и принтеры, **они упорядочивают доступ к данным и принтерам**, что позволяет избежать конфликтов и **сделать именно совместный доступ предсказуемым**.

То есть, эти серверы не просто выполняют обслуживание компьютеров-клиентов, но и управляют своими "общими ресурсами", и устанавливают "правила игры" с ними. Они не являются подчиненными компьютерами. Но не являются и "командирами", так как каждый клиент сам решает, нужно ли ему обращаться к серверу.

К сожалению, невозможно просто подключить большой накопитель просто к сети. Представьте, что один компьютер пытается прочитать данные, которые другой компьютер при этом как раз записывает. Или два компьютера пытаются одновременно изменить одни и те же данные. Это те самые конфликты доступа, которые и позволяет устранить файловый сервер.

Невозможно и просто подключить принтер к сети. Если два компьютера одновременно начнут вывод на печать, на бумаге получится каша. Сервер печати позволяет разрешить такие конфликты. Он получает запросы на печать результатов расчетов и сохраняет их у себя до окончания расчетов клиентами.

При этом он может и сам пользоваться услугами файлового сервера. То есть, сервер может быть одновременно и клиентом другого сервера. А после завершения работы клиента сервер печати выводит на печать сформированный документ как одно целое. В результате документы разных клиентов не перепутываются.

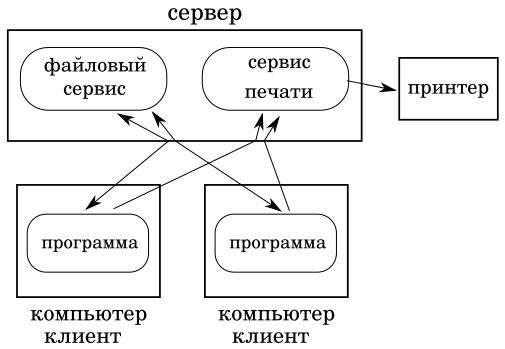
Возникает вопрос, является ли сервер печати посредником по отношению к файловому серверу, если он пользуется его услугами для временного сохранения печатаемых документов? Нет, не является. Так как он действительно сам выполняет услугу "печать документа". Где именно, и как именно, сохраняются промежуточные результаты не имеет никакого значения.

**Компьютер один, но серверов несколько. Или, появляются сервисы**

Разумеется, нет необходимости обязательно выделять разные компьютеры для выполнения функций файлового сервера и сервера печати. Обе функции может выполнять один и тот же компьютер. Вот тут и становится важным нюанс, о которым мы говорили ранее. Взаимодействуют на самом деле не с компьютером, и не компьютеры. Взаимодействуют программы, которые выполняются на компьютерах.

То есть, функции обслуживания, на самом деле, выполняет не компьютер, а именно программа им выполняемая. **Программа-сервер**. И тут возникает некоторая неоднозначность. У нас есть и компьютер-сервер, и выполняемые им программы-серверы. И там, и там, сервер. На самом деле, в этом нет ничего страшного.

Тем не менее, не редко программы-серверы называют сервисами, то есть, услугами. А собственно компьютер при этом остается сервером. Сервер (обслуживающий) оказывает услуги (сервисы).



Один сервер оказывает несколько услуг-сервисов. Иллюстрация моя

На первый взгляд, разница не велика. Один физический сервер, или несколько, большого значения не имеет. Однако, все не так просто.

**Серверы приложений и приложения серверы**

Дальнейшее развитие технологий привело к появлению более сложных серверов. В качестве примера можно привести серверы баз данных. Сервер баз данных уже предоставляет не услугу совместного доступа к данным, а **услугу обработки данных**. И это важный и большой шаг вперед.

Обработка данных это уже функция приложения. И сервер теперь предоставляет услугу **совместного использования приложения**. Это уже гораздо более высокий уровень.

Теперь клиент может запросить не просто доступ к данным, но и некоторую обработку данных. Например, отбор информации на основе некоторых признаков, сортировку, объединение различной информации в единое целое.

Представьте, что у нас нет сервера баз данных, а есть обычный файловый сервер. Если нам нужно выполнить даже просто поиск информации в большой файле, который хранится на сервере, потребуется передать весь файл с сервера клиенту по сети. А это большая нагрузка на сеть, да и таких одновременно работающих клиентов может быть много.

Но если перенести функцию обработки данных на сервер, то потребуется передача только результатов от сервера к клиенту. В большинстве случаев, это гораздо меньший объем информации. А значит, снижается нагрузка а сеть.

Но ничто не бывает бесплатным. Перенос обработки данных на сервер, да еще при необходимости одновременно обрабатывать много таких высокоуровневых запросов, требует использования гораздо более мощного компьютера, с большим объемом памяти. А это совсем не дешево.

Но одновременно, это позволяет и снизить требования к компьютерам-клиентам. А значит, они могут быть более дешевыми. Вместе со снижением нагрузки на сеть, это снижает суммарную стоимость всей системы в целом. Поэтому выделенный сервер, и даже серверы, баз данных весьма распространены.

Но разве услуга обработки данных исчерпывается лишь работой с базами данных? Конечно нет! Ведь можно предоставлять и услуги математических вычислений. И компьютеры клиенты могут обращаться к приложениям, расположенным на сервере, для выполнения расчетов, которые сами они выполнить не могут (например, недостаточно мощные).

Можно предоставлять услуги обработки графических данных, преобразования форматов файлов, шифрования и передачи сообщений, и многого другого.

То есть, у нас появляется множество приложений, которые предоставляют самые разные услуги. И это полноценные приложения-серверы. То есть, эти приложения уже сами представляют разные услуги-сервисы, внутри себя.

Физический, аппаратный, сервер-компьютер выполняет несколько приложений-серверов, которые предоставляют различные услуги-сервисы. Это сервер приложений.

Однако, любой компьютер-клиент, точнее, программа-клиент, все еще знает, что она обращается за обслуживанием к серверу. Такие обращения к серверу присутствуют в программе-клиенте в явном виде.

**Распределенные вычисления**

Но мы можем сделать еще один шаг. Мы можем предоставлять не услуги приложения, а **услуги процедуры**. Например, программе требуется вычисление интеграла для некоторой функции. Предположим, что для это используется вызов процедуры Intergal, который программист включает в текст программы.

В случае обычной программы, процедура Intergral будет выполняться на том же самом компьютере, что и собственно программа. Но, например, средствами операционной системы мы можем перенаправить вызов Intergal на другой компьютер. При этом программа может даже не всегда знать, где именно выполняются вычисления.

То есть, работа программы уже не будет ограничиваться одним компьютером. Программа окажется распределенной между несколькими компьютерами в сети.

И сразу хочется спросить, а где же здесь сервер? А здесь нет выделенного сервера. Процедуры, части программы, требующие выполнения большого объема математических вычислений, будут выполняться на одном компьютере, который и будет для них являться сервером. Обработка графики будет выполняться на другом компьютере, который будет для этого сервером. Обработка большого объема данных переместится на еще один компьютер. И это серверы процедур могут даже работать полностью, или частично, параллельно.

**Так что же такое сервер?**

Мы рассмотрели множество примеров. Но все они лишь подтверждают сказанное еще в самом начале статьи. **Сервер обслуживает клиентов выполняя их запросы**. **Предоставляет услуги клиентам**.

Сервер может быть как аппаратным, так и программным. При этом и аппаратный, и программный сервер может предоставлять сразу несколько услуг. Аппаратные серверы могут быть выделенными для одной, или нескольких услуг. Программные серверы могут выполняться как а аппаратных серверах,включая выделенные, так и компьютерах клиентах.

А еще, сервер может быть одновременно и клиентом для других серверов. И может быть посредником. И все это одновременно.

Выделенного сервера может и не быть. При этом его функции могут быть распределены между компьютерами сети.

При этом сервер не является подчиненным объектом. Как не является и главным объектом. Но, **в пределах своей функции, сервер имеет право устанавливать правила работы**.

**Заключение**

Само понятие сервер действительно является простым. Но реальные серверы, хоть аппаратные, хоть программные, чаще всего являются весьма сложными. И дорогими. Поэтому их и устанавливают, физические сервер-компьютеры, в специальных помещениях.

Они не "управляют миром", но действительно выполняются чрезвычайно важные задачи. И отказа сервера это действительно ЧП.

В заключение приведу небольшой список статей канала, которые тоже могут быть интересны школьникам и читателям, повседневная деятельность которых далека от техники