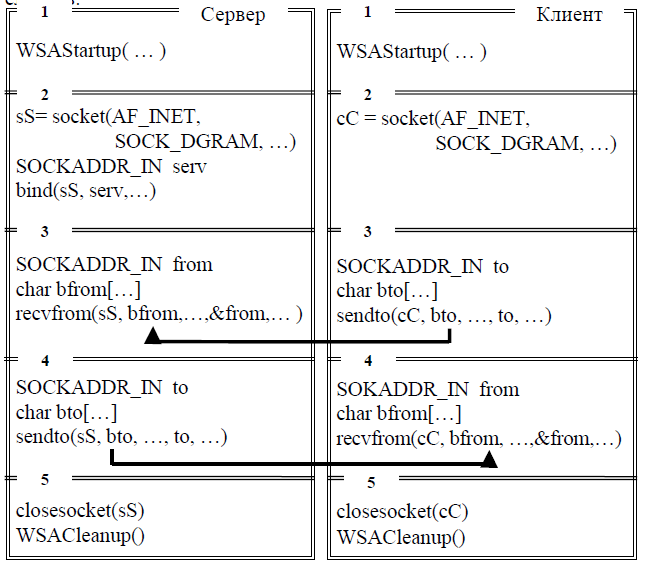
1. **Структура UDP-сервера.**
2. **Структура UDP-клиента.**



Первые блоки обеих программ одинаковые и предназначены для инициализации библиотеки WS2\_32.dll. Второй блок программы-сервера создает сокет (функция socket) и устанавливает его параметры. Следует обратить внимание на параметр SOCK\_DGRAM функции socket, указывающий на тип сокета (в данном случае – сокет, ориентированный на сообщения). Для установки параметров сокета используется функция bind. При этом говорят, что сокеты связывают параметрами. Для хранения параметров сокета в Winsock2 предусмотрена специальная структура SOCKADDR\_IN. перед выполнением функции bind, которая использует эту структуру в качестве параметра, необходимо ее заполнить данными. В SOCKADDR\_IN хранится IP-адрес и номер порта сервера.

В третьем блоке программы сервера выполняется функция recvfrom, которая переводит программу сервера в состояние ожидания до поступления сообщения от программы клиента (функция sendto). Функция recvfrom тоже использует структуру SOCKADDR\_IN. В нее автоматически помещаются параметры сокета клиента после приема от него сообщения. Данные поступают в буфер, который обеспечивает принимающая сторона (на рисунке символьный массив bfrom). Следует отметить, что в качестве параметра функции recvfrom используется связанный сокет и именно через него осуществляется передача данных.

Четвертый блок программы сервера предназначен для пересылки данных клиенту. Процесс осуществляется с помощью функции sendto. В качестве параметров sendto использует структуру SOCKADDR\_IN с параметрами сокета принимающей стороны (в данном случае клиента) и заполненный буфер с данными.

Пятые блоки программ сервера и клиента одинаковые и предназначены для закрытия сокета и завершения работы с библиотекой WS2\_32.dll.

Всем блокам программы клиента, кроме второго, есть аналог в программе сервера. Второй блок по сравнению с сервером не использует команду bind. Здесь проявляется основное отличие между сервером и клиентом. Если сервер должен использовать однозначно определенные параметры (IP-адрес и номер порта), то для клиента это не обязательно – ему Windows выделяет эфемерный порт. Поскольку инициатором связи является клиент, то он должен точно «знать» параметры сокета сервера, а свои параметры клиент получит от Windows и сообщит их вместе с переданным пакетом серверу.

1. **Отличие UDP-сервера от TCP-сервера.**

Основным отличием между UDP-сервером и TCP-сервером является протокол, используемый для передачи данных.

TCP (Transmission Control Protocol) - это протокол, который обеспечивает надежную передачу данных между клиентом и сервером. Он используется для передачи данных, которые должны быть доставлены без потерь и в правильном порядке. При использовании TCP каждое сообщение проверяется на наличие ошибок и повторяется, если были потери данных. Это означает, что TCP-сервер обеспечивает более высокий уровень надежности и точности в передаче данных, чем UDP-сервер.

UDP (User Datagram Protocol) - это протокол, который не гарантирует доставку пакетов данных в правильном порядке и не проверяет на наличие ошибок. Он используется для передачи данных, которые не так важны и могут быть потеряны без серьезных последствий. При использовании UDP-сервера данные передаются непосредственно между клиентом и сервером, и сервер не обязан подтверждать получение сообщения. Это делает UDP-серверы более быстрыми и менее надежными, чем TCP-серверы.

Таким образом, при выборе протокола для своего сервера необходимо учитывать требования к скорости и надежности передачи данных. Если для вашего приложения критична точность и надежность передачи данных, следует использовать TCP-протокол. Если же вы склоняетесь к быстрому передачи данных, и при этом готовы принять потери и ошибки, то можно использовать UDP-протокол.

1. **Отличие UDP-клиента от TCP-клиента.**

Основное отличие между UDP-клиентом и TCP-клиентом заключается в способе установления соединения и передачи данных.

TCP-клиент устанавливает надежное двустороннее соединение с TCP-сервером перед передачей данных. Передача данных происходит в определенной последовательности, и клиент получает подтверждение от сервера об успешной доставке каждого пакета данных. Если пакет данных не был доставлен, TCP-протокол повторяет его передачу, пока пакет не будет доставлен или не будет исчерпано максимальное количество попыток. Это делает TCP надежным, но может приводить к задержкам, особенно если сеть находится под нагрузкой.

UDP-клиент не устанавливает соединения с UDP-сервером и не получает подтверждения о доставке пакетов. Вместо этого UDP-клиент просто отправляет пакеты данных на UDP-сервер и надеется, что они будут успешно доставлены. Если пакет был потерян, то UDP-протокол не будет повторять его передачу, и клиент не получит никаких уведомлений об ошибке. Это делает UDP ненадежным, но более быстрым и подходящим для передачи потоковых данных, таких как аудио и видео.

1. **Что такое канал связи с точки зрения TCP?**

Канал связи (или соединение) создается между двумя сокетами, ориентированными на поток. На стороне сервера это должен быть связанный (функция bind) и переключенный в режим прослушивания (функция listen) сокет. На стороне клиента должен быть создан дескриптор ориентированного на поток сокета (функция socket).

Канал связи создается в результате взаимодействия функция accept (на стороне сервера) и connect (на стороне клиента). Алгоритм взаимодействия этих функция зависит от установленного режима ввода-вывода для участвующих в создании каналов сокетов.

1. **Что значит: обмен без соединения (ориентированный на сообщения)?**

Это такой обмен данными (сообщениями), особенностью которого является то, что протоколом, который обеспечивает такой обмен, не гарантируется доставка и правильная последовательность приема отправленных сообщений. Весь контроль надежности доставки сообщений возлагается на разработчика приложения. В связи с этим, обмен данными с помощью сообщений используется в основном для широковещательных сообщений или для пересылки коротких сообщений, последовательность получения которых не имеет значения.

1. **Что значит UDP ненадежный протокол?**

Ненадежный протокол – это такой протокол, который обладает следующими свойствами:

* отсутствует механическое обеспечение надежности: пакеты не упорядочиваются, и их прием не подтверждается;
* отсутствует гарантия доставки;
* отсутствует обработка соединений
* невозможно вычислить контрольную сумму для пакета данных
* отсутствие буферизации
* нельзя разбивать сообщения на несколько пакетов

1. **В каких случаях следует выбирать UDP-протокол для обмена данными?**

Обмен данными с помощью сообщений используется в основном для широковещательных сообщений или для пересылки коротких сообщений, последовательность получения которых не имеет значения.

1. **Что значит: UDP не буферизирует соединения?**

Обрабатывает только один пакет, который пришел. Пока не обработает этот пакет, другие пакеты не обрабатываются, т.к. нет буфера памяти.

1. **Можно ли одновременно задействовать один номер порта для двух серверов: TCP и TCP, UDP и UDP, TCP и UDP? если возможно, то продемонстрируйте.**

Уникальность по трем параметрам:

1. IP
2. Порт
3. Протолок

Если хоть что-то отличается то мона.

1. **Определите порт UDP-сервера с помощью утилиты NETSTAT.**

0.0.0.0:2000 (порт 2000)

1. **Продемонстрируйте потерю сообщений UDP-сервером. Используйте функцию Sleep.**