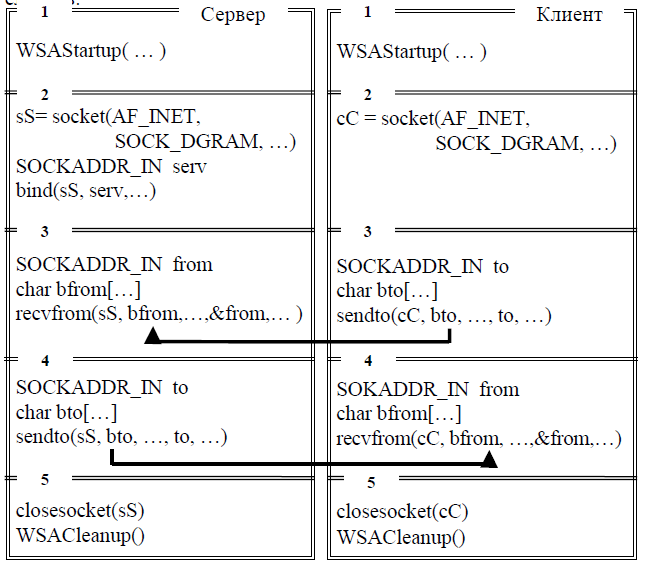
1. **Структура UDP-сервера.**
2. **Структура UDP-клиента.**



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.**

UDP (User Datagram Protocol) и TCP (Transmission Control Protocol) - это два протокола передачи данных, которые используются в сетевых приложениях для установления соединения и передачи данных между узлами сети.

Основное отличие между UDP-сервером и TCP-сервером заключается в том, как они управляют передачей данных:

1. Надежность передачи: TCP-сервер обеспечивает надежную передачу данных, т.е. гарантирует, что все отправленные данные будут доставлены получателю в правильном порядке и без потерь. В то время как UDP-сервер не гарантирует надежную доставку данных, и потери пакетов могут произойти в случае перегрузки сети или других факторов.
2. Установление соединения: TCP-сервер устанавливает соединение с клиентом перед началом передачи данных, т.е. сначала происходит процесс установления соединения, а затем передача данных. В отличие от TCP, UDP-сервер не устанавливает соединение и начинает передачу данных немедленно.
3. Производительность: UDP-сервер более быстрый, чем TCP-сервер, так как не требует установления соединения и проверки надежности передачи данных. Однако это может привести к потере данных или их дублированию, если сеть перегружена или ненадежна.
4. Размер пакета: TCP-пакеты обычно более объемные, чем UDP-пакеты, так как TCP включает в себя много служебной информации для обеспечения надежности передачи данных.
5. Применение: UDP-серверы часто используются для передачи потоковых данных, таких как аудио или видео, где некоторая потеря данных не столь критична. В то время как TCP-серверы наиболее подходят для передачи данных, где требуется высокая степень надежности и точности
6. Отличие UDP-клиента от TCP-клиента.
7. Что такое канал связи с точки зрения TCP?
8. Что значит: обмен без соединения(ориентированный на сообщения)?
9. Что значит UDP ненадежный протокол?
10. В каких случаях следует выбирать UDP-протокол для обмена данными?
11. Что значит: UDP не буферизирует соединения?
12. Можно ли одновременно задействовать один номер порта для двух серверов: TCP и TCP, UDP и UDP, TCP и UDP? Если возможно, то продемонстрируйте.
13. Определите порт UDP-сервера с помощью утилиты NETSTAT.
14. Продемонстрируйте потерю сообщений UDP-сервером. Используйте функцию Sleep.