

Технологии программирования

Курсовая работа

Часть №4

Тема: Сокеты Беркли

Задание: Разработать сетевое приложение, позволяющее передавать текстовые сообщения в масштабах локальной сети. Все варианты реализации должны быть совместимы между собой (использовать один протокол).

Сокеты Беркли — это стандартный интерфейс программирования межпроцессного и сетевого взаимодействия, разработанный в Калифорнийском университете в Беркли. Интерфейс предназначен для обеспечения максимально легкой портируемости приложений на уровне исходного кода и поддерживается большинством существующих операционных систем. Интерфейс определяет некоторый базовый набор функций, предназначенных в основном для работы в сетях TCP/IP.

Интерфейс является процедурным (как и большинство системных API), состоит из набора функций обслуживания сокета и вспомогательных подпрограмм.

Перед использованием библиотеку сокетов необходимо загрузить и проинициализировать, вызвав соответствующую функцию.

Для создания сокета используется функция `socket`. В качестве аргументов ей передается тип создаваемого сокета (протокол транспортного уровня) и семейство адресов (протокол сетевого уровня). Возвращается хэндл сокета, для дальнейшего использования. Некоторые виды сокетов могут использоваться многократно. Для уничтожения сокета и освобождения связанных с ним ресурсов используется функция `closesocket`.

Сокеты делятся по типу используемого протокола транспортного уровня на потоковые и датаграммные. Первые работают с протоколом TCP, с потоковым механизмом и установкой виртуального соединения, вторые работают с протоколом UDP, механизмом отдельных сообщений (датаграмм). От выбора типа сокета зависит набор используемых для работы с ним функций.

Также сокеты по характеру использования делятся на клиентские и серверные. Клиентские сокеты служат для передачи данных, они инкапсулируют виртуальное соединение. Серверные сокеты принимают запросы на подключение, приходящие по сети.

Рассмотрим основные функции для работы с сокетами.

Функция `socket`. Устанавливает виртуальное соединение для сокетов потокового типа. Без установки соединения работа такого сокета невозможна. Для сокетов датаграммного типа устанавливает адрес назначения «по-умолчанию».

Функция `bind`. Устанавливает локальный адрес сокета. Необходима для инициализации серверного сокета — задает адрес и порт, на котором сокет будет ожидать соединения. Для клиентских сокетов задает адрес и порт отправителя. Если в клиентском соquete эта функция не была использована, адрес и порт отправителя назначаются автоматически операционной системой. Обратите внимание, что использование этой функцией

основного диапазона портов (обычно до 1024) требует, чтобы приложение было запущено с привилегиями администратора.

Функция `listen`. Переключает серверный сокет в «слушающий» режим. Именно после вызова этой функции сокет и становится серверным — т.е. способным принимать соединения.

Функция `accept`. Принимает входящее соединение к серверному сокету. Возвращает адрес и порт отправителя для его идентификации. Самое главное — эта функция устанавливает логическое соединение со стороны сервера и создает клиентский сокет для работы с данным подключением. Все взаимодействие с клиентом ведется через созданный клиентский сокет. Передача данных через серверный сокет невозможна.

Функция `recv`. Считывает данные из сокета.

Функция `send`. Отправляет данные в сокет.

Функция `recvfrom`. Считывает данные и адрес отправителя. Применима только для датаграммных сокетов.

Функция `sendto`. Отправляет данные конкретному получателю. Применима только для датаграммных сокетов.

Датаграммные сокеты, поскольку они не требуют установки виртуального соединения, можно использовать и как серверные и как клиентские одновременно.

Функция `setsockopt`. Устанавливает параметры сокета.

Функция `ioctlsocket`. Устанавливает параметры сокета.

Вспомогательные функции

`hton*` преобразуют порядок байт из машинного в сетевой. В случае x86 меняют порядок байт на противоположный (сетевой порядок `big-endian`, машинный `little-endian`). В случае `big-endian` машин не делают ничего.

`ntoh*` обратные функции, преобразуют порядок байт из сетевого в машинный.

Функция `gethostbyname`. Получить адрес (IP) по имени узла. Используются все доступные ОС службы — DNS, WINS, SAMBA, hosts, AD и прочие.

Функция `gethostbyaddr`. Получить имя узла по его адресу.

Для выполнения работы всем студентам группы необходимо составить единый протокол взаимодействия для обеспечения совместимости между различными реализациями. Вы можете выбрать любой доступный вариант реализации, использовать любые сокеты в любом количестве.

Необходимо предусмотреть вывод списка участников, отправку сообщений всем и конкретному участнику (приват).