Интерфейс сокетов Windows

Происхождение Winsock

- Интерфейс Беркли — всего лишь одна (хотя и чаще всего используемая) реализация интерфейса прикладного программирования, основанная на модели сокетов.
- Сокеты Windows (часто называемые “Winsock”) — также интерфейс прикладного программирования, разработанный на основе сокетов Беркли. Тогда как сокеты Беркли используются на разных операционных системах, Winsock предназначен исключительно для семейства Windows.
- В состав Winsock входит множество функций из интерфейса Беркли, изначально разработанных для Unix — операционной системы, для которой и предназначались сокеты Беркли. Кроме того, имеется набор специфических для Windows функций, позволяющих программистам пользоваться преимуществами интерфейса Windows, основанного на передаче сообщений.

Реализация Winsock

- Спецификация Windows Sockets описывает стандарт, по которому программы Windows обязаны общаться с сетями на базе TCP/IP. Таким образом, Windows Sockets API представляет колоссальные возможности в развитии старых и написании новых сетевых приложений. Корпорация Microsoft, создатель и владелец Windows, не имеет имущественных прав на стандарт Winsock. Усилия, необходимые для создания спецификации Windows Sockets, были затрачены большим количеством людей, работающих в различных корпорациях. Цель разработки — создание единого интерфейса прикладного программирования (API), который бы использовался в равной мере как разработчиками, так и продавцами сетевого программного обеспечения и служил бы стандартом работы с TCP/IP для Windows.
- Интерфейс сокетов Windows не входит в состав Windows, а реализован в виде динамически загружаемой библиотеки (DLL).

Краткое обозрение библиотеки функций Winsock

- Интерфейс прикладного программирования Winsock содержит набор (библиотеку) функций общего пользования, требуемых для решения определенного класса задач. Спецификация Winsock разделяет всю библиотеку на три группы:
  - Функции сокетов в стиле Беркли, включенные в состав Winsock API.
  - Функции для работы с базами данных, позволяющие программам получать информацию об именах доменов, коммуникационных службах и протоколах.
  - Специфические функции Windows, расширяющие набор функций интерфейса сокетов Беркли.

Функции интерфейса сокетов

- Блокирующая операция задерживает выполнение программы до окончания своей работы. Как правило, все функции сетевого ввода-вывода сокетов в стиле Беркли — блокирующие. На деле, блокирование проявляется задержкой в выполнении программы до окончания передачи-приема сетевых данных.

Функции сокетов в стиле Беркли, которые могут блокировать выполнение операций в Winsock API.

Accept

Подтверждает запрос на установление соединения. Образует новый сокет и соединяет его с удаленным сетевым компьютером, запрашивающим соединение. Исходный сокет возвращается в состояние приема входящих запросов.

closesocket

- Закрывает одну сторону в соединении сокетов.

connect

- Устанавливает соединение на указанном сокете.

recv

- Принимает данные из соединенного сокета.

recvfrom

- Принимает данные из соединенного или не соединенного сокета.

select

- Выполняет синхронные мультиплексные операции ввода-вывода путем наблюдения за состоянием нескольких сокетов.

send

- Передает данные через соединенный сокет.

sendto

- Передает данные через не соединенный или соединенный сокет.

- Можно заметить, что все, описанные выше, функции либо производят операции ввода-вывода, либо ждут окончания сетевого ввода-вывода до того, как завершить выполнение. Отсюда можно сделать вывод, что любая функция, так или иначе связанная с операциями ввода-вывода, может блокировать выполнение остальных функция Winsock API.

- Следующие функции не выполняют операций ввода-вывода в процессе работы. Они либо преобразуют информацию, либо имеют дело с локальным сокетом сетевого компьютера. Другими словами, их деятельность не связана с удаленными сетевыми устройствами. Поэтому, несмотря на то, что они также являются функциями в стиле Беркли, ни одна из них не блокирует операции прикладной программы.

Функции в стиле Беркли, не блокирующие работу Winsock API

bind

- Присваивает имя неинициализированному (новому) сокету.

getpeername

- Получает имя удаленного процесса, связанного с указанным сокетом. (Winsock API хранит эту информацию в локальной структуре данных, поэтому вызов данной функции не связан с операциями сетевого ввода-вывода.)

getsockname

- Возвращает имя указанного местного (локального) сокета.

getsockopt

- Возвращает статус (опции) указанного сокета.

htonl

- Преобразует порядок байтов в 32-разрядном числе из машиннозависимого в сетевой.

htons

- Преобразует порядок байтов в 16-разрядном числе из машиннозависимого в сетевой.

inetaddr

- Преобразует строку с IP-адресом в формате десятичное с точкой в 32-разрядное двоичное число (с сетевым порядком байтов).

inetntoa

- Преобразует IP-адрес в формат десятичное с точкой.

ioctlsocket

- Управляет параметрами сокета, относящимися к обработке операций сетевого ввода/вывода.

listen

- Переводит указанный сокет в состояние прослушивания запросов на входное соединение. (Функция переводит сокет в режим прослушивания, однако сама по себе не производит никаких операций сетевого ввода-вывода.)

ntohl

- Преобразует порядок байтов 32-разрядного числа из сетевого в машиннозависимый (порядок хоста).

ntohs

- Преобразует порядок байтов 16-разрядного числа из сетевого в машиннозависимый (порядок хоста).

setsockopt

- Устанавливает режим (опции) работы сокета.

shutdown

- Закрывает одну сторону дуплексного соединения (только для местного компьютера).

socket

- Образует точку сетевого соединения и возвращает дескриптор сокета.

Функции Winsock API для работы с базами данных

- Они позволяют прикладной программе получать информацию об именах доменов, коммуникационных службах и протоколах. Для выполнения своих задач этим функциям приходится обращаться к разнообразным, как местным, так и удаленным, источникам информации. Спецификация Winsock определяет интерфейс, а также формат данных, который возвращается функциями для работы с базами данных. Вопросы хранения и выборки данных зависят от конкретной реализации Winsock и не оговариваются в стандарте.

gethostbyaddr

- Возвращает имя (имена) по указанному сетевому адресу.

gethostbyname

- Возвращает имя (имена) и IP-адрес, соответствующие указанному сетевому имени.

gethostname

- Возвращает имя локального сетевого компьютера.

getprotobyname

- Возвращает официальное имя и номер протокола по указанному имени (например, TCP). (В семействе протоколов TCP/IP каждому протоколу соответствует уникальное целое число.)

getprotobynumber

- Возвращает имя и номер протокола по указанному номеру.

getservbyname

- Возвращает имя сетевой службы (например, time) и номер порта протокола, соответствующие указанному имени.

getservbyport

- Возвращает имя сетевой службы и номер порта протокола, соответствующие указанному номеру порта.

- Некоторые из вышеописанных функций возвращают указатели на значения, хранящиеся в системной области памяти. Реализация Winsock может использовать такие области повторно при вызове следующей функции. Это значит, что, если программа желает использовать полученное значение переменной, его необходимо переписать в переменную, принадлежащую программе, до следующего вызова функций, иначе оно может оказаться утерянным. Данные, находящиеся в системных буферах, действительны только до следующего вызова функции Winsock.

- Интерфейс прикладного программирования Winsock включает так называемые асинхронные (специфические для Windows) аналоги всех функций для работы с базами данных, за исключением gethostname. В дальнейшем мы еще обсудим асинхронные функции Windows. Асинхронные функции представляют собой Windows-расширения интерфейса сокетов Беркли. Они появились в результате стремления разработчиков использовать встроенный в Windows механизм обмена сообщениями.

Асинхронные функции для работы с базами данных в составе Winsock API

Функция в стиле Беркли	Асинхронный эквивалент
gethostbyaddr	WSAAsyncGetHostByAddr
gethostbyname	WSAAsyncGetHostByName
gethostname	Не имеет эквивалента
getprotobyname	WSAAsyncGetProtoByName
getprotobynumber	WSAAsyncGetProtoByNumber
getservbyname	WSAAsyncGetServByName
getservbyport	WSAAsyncGetServByPort

Функции-расширения, специфические для Windows

WSAAsyncSelect

- Асинхронный вариант функции select

WSACancelAsyncRequest

- Прекращает некорректное выполнение функции WSAAsyncGetXByY.

WSACancelBlockingCall

- Прекращает некорректное выполнение блокирующей функции API.

WSACleanup

- Уведомляет Windows Sockets о том, что программа закончила работу с DLL.

WSAGetLastError

- Возвращает сообщение о последней ошибке при вызове функции.

WSAIsBlocking

- Определяет, является ли низлежащий уровень Winsock DLL блокирующим.

WSASetBlockingHook

- Устанавливает ловушку блокирующего вызова.

WSASetLastError

- Фиксирует сообщение об ошибке для последующего вызова WSAGetLastError.

WSAStartup

- Инициализирует низлежащий уровень Winsock DLL.

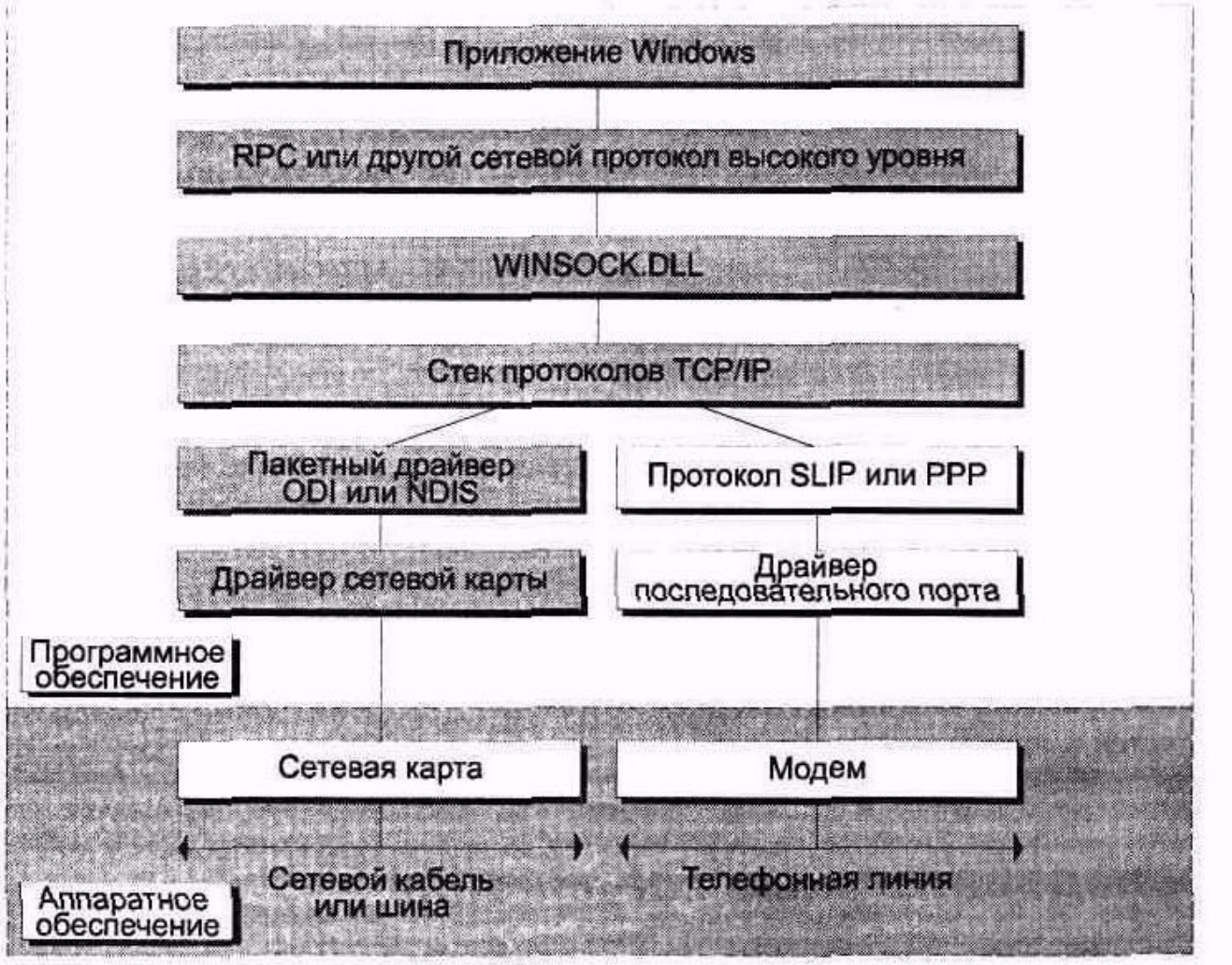
WSAUnhookBlockingHook

- Восстанавливает первоначальную блокирующую функцию.

Общая картина

- Чтобы осознать картину программирования в Windows целиком, нужно понимать, какое место в ней занимает Winsock и какие именно функции системы им выполняются. Множество фирм продают свои приложения, полезные утилиты и программы Для того кто не знаком со структурой Winsock, бывает сложно определить, где заканчивается разработка и начинается Winsock.
- Модуль WINSOCK.DLL находится между стеком протоколов TCP/IP и клиентскими приложениями. Другими словами, он управляет интерфейсом к стеку TCP/IP.

интерфейс Winsock вписывается в общую схему разработки TCP/IP программ для Windows.



Протоколы высокого уровня

- Как правило, интерфейс протоколов высокого уровня, показанный на рис., разрабатывается поставщиком программного продукта. Этот уровень располагается между программами Windows и модулем WINSOCK.DLL. Протоколом высокого уровня может являться продукт *вызова удаленных процедур* (Remote Procedure Call, RPC) или что-нибудь еще, обеспечивающее подобные услуги.

Создание нового сокета

- Следующий оператор демонстрирует, как создается новый сокет:
- **socket_handle = socket(protocol_family, socket_type, protocol);**
- Дескриптор сокета Winsock не соответствует дескриптору сокета Беркли. Все аргументы, однако, в точности соответствуют аргументам функции в стиле Беркли.

Настройка сокета

- Используются те же функции что и для сокетов Беркли.

Соединение через сокет

- Следующий оператор демонстрирует образец вызова connect:
- **result = connect(socket_handle, remote_socket_address, address_length);**

- **Другие функции так же аналогичны**

Файлы заголовков

- В программах Winsock указывается только один файл-заголовок winsock.h. Он указывается всегда (#include <winsock.h>), если приложение работает с сокетами или вызывает любую функцию, входящую в Winsock. В большинстве Windows-программ требуется также указывать файл-заголовок windows.h. Если winsock.h включен, то необходимости дополнительно включать windows.h нет — это делается автоматически в файле winsock.h, поскольку он не может обойтись без windows.h.

Управление данными

- При разработке Windows-программ необходимо учитывать следующую особенность: любой объект памяти, например буфер или переменная, должен быть доступен для WINSOCK. DLL в любой момент времени при исполнении операций Winsock. В многопоточных версиях Windows программа обязана самостоятельно координировать доступ к объектам памяти. Об этом прямо сказано в спецификации Winsock, где указано, что реализации Winsock (WINSOCK.DLL) не отвечают за управление памятью программы.

Обязательные функции Winsock

- В Winsock определены две обязательные для исполнения функции: WSAStartup и WSACleanup. Первая вызывается перед тем, как вызвать любую другую функцию Winsock. Вторая — по окончании работы с Winsock. Для каждой функции WSAStartup всегда должна вызываться соответствующая функция WSACleanup.

Функция WSAStartup

- Функция WSAStartup позволяет программе указать требуемую версию Winsock. Она также позволяет программе получить информацию о конкретной реализации Winsock. Фактически, при вызове WSAStartup происходит диалог между программой и модулем WINSOCK.DLL. Возможность узнавать версию Winsock предусмотрена для того, чтобы обеспечить разработку программ, использующих более новые и не работающих со старыми версиями Winsock. Такая программа указывает минимальную версию, с которой она еще может работать, а Winsock — наиболее высокую версию из тех, что он может обеспечить. Далее программа решает, как ей лучше поступить в том или ином случае.

Функция WSACleanup

- Прикладная программа может вызывать WSAStartup несколько раз за время работы. Это может понадобиться, например, если в разных частях программы требуется узнать версию WINSOCK.DLL. И вместо того чтобы отводить глобальный массив для хранения версии, программа вызывает WSAStartup. Функция WSACleanup органично дополняет WSAStartup. На каждый вызов функции WSAStartup должен приходиться вызов WSACleanup. Winsock регистрирует каждый вызов WSAStartup при помощи внутреннего счетчика вызовов. Каждый раз при вызове WSAStartup значение счетчика увеличивается. При вызове WSACleanup, наоборот, уменьшается. Последняя, вызванная программой функция WSACleanup (устанавливающая счетчик в ноль) заставляет Winsock произвести операции по очистке буферов, переменных и т. д., имевших отношение к программе.

- Реализация Winsock (WINSOCK.DLL) отводит внутренние статические области памяти для зарегистрировавшейся в ней программы. Каждая программа получает собственные области памяти, к которым можно обращаться только через Winsock API. Области памяти являются внутренними для Winsock точно так же, как и структуры данных сокета. Последняя вызванная WSACleanup говорит Winsock, что он не нужен больше программе, следовательно, можно освободить все ресурсы, отведенные ранее для этой программы. Функция WSAStartup начинает работу программы с Winsock, а WSACleanup прекращает ее.

- Как только WSACleanup вызвана в последний раз, Winsock отсоединяет все оставшиеся сокеты, ориентированные на соединение. Тем не менее все данные, оставшиеся в выходных очередях, отправляются. Предположим, что закрытие сокета следует сразу же за командой send. Даже если сеть и не успела передать все данные до вызова WSACleanup, они все равно будут посланы Winsock.