**. Обмен данными без соединения**

Если для передачи данных на транспортном уровне используется протокол UDP, то говорят, об обмене данными без соединения или об обмене данными с помощью сообщений. Для отправки и приема сообщений в Winsock2 используются функции sendto и recvfrom (рисунки 3.12.1 и 3.12.2). При этом предполагается, что сообщения будут курсировать между сокетами ориентированными на сообщения.

Особенностью использования этих функций заключается в том, что протоколом UDP не гарантируется доставка и правильная последовательность приема отправленных сообщений. Весь контроль надежности доставки и правильной последовательностью поступления 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.

// **-- отправить сообщение**

// **Назначение:** функция предназначена для отправки сообщения

// без установления соединения

//

**int sendto(**

**SOCKET s,** // [in] дескриптор сокета

**const char\* buf,** // [in] буфер для пересылаемых данных

**int len,** // [in] размер буфера **buf**

**int flags,** // [in] индикатор режима маршрутизации

**const struct sockaddr\* to,** // [in] указатель на **SOCKADDR\_IN**

**int tolen** // [in] длина структуры to

**);**

// **Код возврата:** при успешном завершении функция возвращает

// количество пересланных байт данных, иначе

// возвращается **SOCKET\_ERROR**

// **Примечания:** - функция может применяться только для

// сокетов, ориентированных на сообщения

// (**SOCK\_DGRAM**))

// - параметр **to** указывает на структуру

// **SOCKADDR\_IN** с параметрами сокета получателя;

// - для параметра **flags** рекомендуется установить

// значение **NULL**

Рисунок 3.12.1. Функция sendto

Также как и функции send и recv, функции sendto и recvfrom работают в синхронном режиме, т.е. вызвав, например, функцию recvfrom вызывающая программа не получит управления до момента завершения приема данных.

Следует также обратить внимание, что обе функции используют в качестве параметров структуру SOCKADDR\_IN. В случае выполнения функции send, структура должна содержать параметры сокета получателя. У функция recv структура SOCKADDR\_IN, наоборот, предназначена для получения параметров сокета отправителя.

Часто протокол UDP (и соответственно функции sendto и recvfrom) используется для пересылки сообщений предназначенных для рассылки одного сообщения всем компьютерам сети (широковещательные сообщения). Для этого в параметрах сокета отправляющей стороны используются специальные широковещательные и групповые IP-адреса. Использование и групповых адресов функцией sendto по умолчанию запрещено. Разрешение

// **-- принять сообщение**

// **Назначение:** функция предназначена для получения сообщения

// без установления соединения

**int recvfrom(**

**SOCKET s,** // [in] дескриптор сокета

**char\* buf,** // [out] буфер для получаемых данных

**int len,** // [in] размер буфера **buf**

**int flags,** // [in] индикатор режима маршрутизации

**struct sockaddr\* to,** // [out] указатель на **SOCKADDR\_IN**

**int\* tolen** //[out] указатель на размер to

**);**

// **Код возврата:** при успешном завершении функция возвращает

// количество принятых байт данных, иначе

// возвращается **SOCKET\_ERROR**

// **Примечания:** - функция может применяться только для

// сокетов, ориентированных на сообщения

// (**SOCK\_DGRAM**));

// - параметр **to** указывает на структуру

// **SOCKADDR\_IN** с параметрами сокета отправителя;

// - параметр **tolen** содержит адрес четырех байт, с

// размером структуры **SOCADDR\_IN**

// - для параметра **flags** рекомендуется установить

// значение **NULL**

Рисунок 3.12.2 Функция recvfrom

//........................................................................

**try**

**{** //...**WSAStartup(...),sS = socket(...,SOCKET\_DGRAM,...)**

**SOCKADDR\_IN serv**; // параметры сокета sS

**serv.sin\_family = AF\_INET;** // используется IP-адресация

**serv.sin\_port = htons(2000);** // порт 2000

**serv.sin\_addr.s\_addr = INADDR\_ANY;** // адрес сервера

**if(bind(sS,(LPSOCKADDR)&serv, sizeof(serv))== SOCKET\_ERROR)**

**throw SetErrorMsgText("bind:", WSAGetLastError());**

**SOCKADDR\_IN clnt;** // параметры сокета клиента

**memset(&clnt,0,sizeof(clnt));** // обнулить память

**int lc = sizeof(clnt);**

**char ibuf[50];** //буфер ввода

**int lb = 0;** //количество принятых байт

**if (lb = recvfrom(sS, ibuf, sizeof(ibuf), NULL,**

**(sockaddr\*)&clnt, &lc)) == SOCKET\_ERROR)**

**throw SetErrorMsgText("recv:",WSAGetLastError());**

//................................................................

**}**

**catch (string errorMsgText)**

**{cout << endl << errorMsgText;}**

//........................................................................

Рисунок 3.12.3 Пример использования функции recvfrom в программе сервера

на использование широковещательных устанавливается функцией setsockopt.

На рисунках 3.12.3 и 3.12.4 приведены примеры применения функций recvfrom и sendto, которые используются в программах сервера и клиента соответственно.

//........................................................................

**try**

**{**//...**WSAStartup(...),сС = socket(...,SOCKET\_DGRAM,...)**

**SOCKADDR\_IN serv;** // параметры сокета сервера

**serv.sin\_family = AF\_INET;** // используется ip-адресация

**serv.sin\_port = htons(2000);** // порт 2000

**serv.sin\_addr.s\_addr = inet\_addr("127.0.0.1");** // адрес сервера

**char obuf[50]= "client: I here";** //буфер вывода

**int lobuf = 0;** //количество отправленных

**if ((lobuf = sendto(cC,obuf,strlen(obuf)+1,NULL,**

**(sockaddr\*)&serv, sizeof(serv))) == SOCKET\_ERROR)**

**throw SetErrorMsgText("recv:",WSAGetLastError());**

//............................................................

**}**

**catch (string errorMsgText)**

**{cout << endl << errorMsgText;}**

//........................................................................

Рисунок 3.12.3 Пример использования функции sendto в программе клиента