

Глава 4. Интерфейс Named Pipe

4.1. Предисловие к главе

Современные операционные системы имеют встроенные механизмы межпроцессного взаимодействия - *IPC (InterProcess Communication)* [15], предназначенные для обмена данными между процессами и для синхронизации процессов.

Разработчики программного обеспечения могут использовать IPC с помощью предоставляемых операционными системами программных интерфейсов (API). С программными интерфейсами IPC операционной системы Windows можно ознакомиться, например, в [4].

В этой главе рассматривается программный интерфейс одного из IPC-механизмов операционной системы Windows, который может быть использован для обмена данными между распределенными в локальной сети процессами, и имеет название *Named Pipe (именованный канал)*.

4.2. Назначение и состав интерфейса Named Pipe

Именованным каналом называется объект ядра операционной системы, который обеспечивает обмен данными между процессами, выполняющимися на компьютерах в одной локальной сети. Процесс, создающий именованный канал, называется *сервером именованного канала*. Процессы, которые связываются с именованным каналом, называются *клиентами именованного канала*. Любой именованный канал идентифицируется своим именем, которое задается при создании канала.

Именованные каналы бывают: *дуплексные* (позволяющие передавать данные в обе стороны) и *полудуплексные* (позволяющие передавать данные только в одну сторону). Передача данных в именованном канале может осуществляться как потоком, так и сообщениями. Обмен данными в канале может быть *синхронным* и *асинхронным*.

Для использования функций интерфейса Named Pipe в программе на языке C++ необходимо включить в ее текст заголовочный файл Windows.h. Сами функции интерфейса располагаются в библиотеке KERNEL32.DLL ядра операционной системы.

В таблице 4.2.1 перечислены основные функции интерфейса Named Pipe. Следует отметить, что функции CreateFile, ReadFile, WriteFile, которые тоже перечислены в таблице 4.2.1, применяются не только для работы с именованными каналами, но и для работы с файловой системой, с сокетами и т.д. Поэтому эти универсальные функции часто не указывают в составе Named Pipe API.

Все функции Named Pipe API можно разбить на три группы: функции управления каналом (создать канал, соединить сервер с каналом, открыть канал, получить информацию об именованном канале, получить состояние канала, изменить характеристики канала); функции обмена данными (писать в канал, читать из канала, копировать данные канала) и функции для работы с транзакциями.

Таблица 4.2.1

Наименование функции	Назначение
CallNamedPipe	Выполнить одну транзакцию
ConnectNamedPipe	Соединить сервер с каналом
CreateFile	Открыть канал
CreateNamedPipe	Создать именованный канал
DisconnectNamedPipe	Закончить обмен данными
GetNamedPipeHandleState	Получить состояние канала
GetNamedPipeInfo	Получить информацию об именованном канале
PeekNamedPipe	Копировать данные канала
ReadFile	Читать данные из канала
SetNamedPipeHandleState	Изменить характеристики канала
TransactNamedPipe	Писать и читать данные канала
WaitNamedPipe	Определить доступность канала
WriteFile	Писать данные в канал

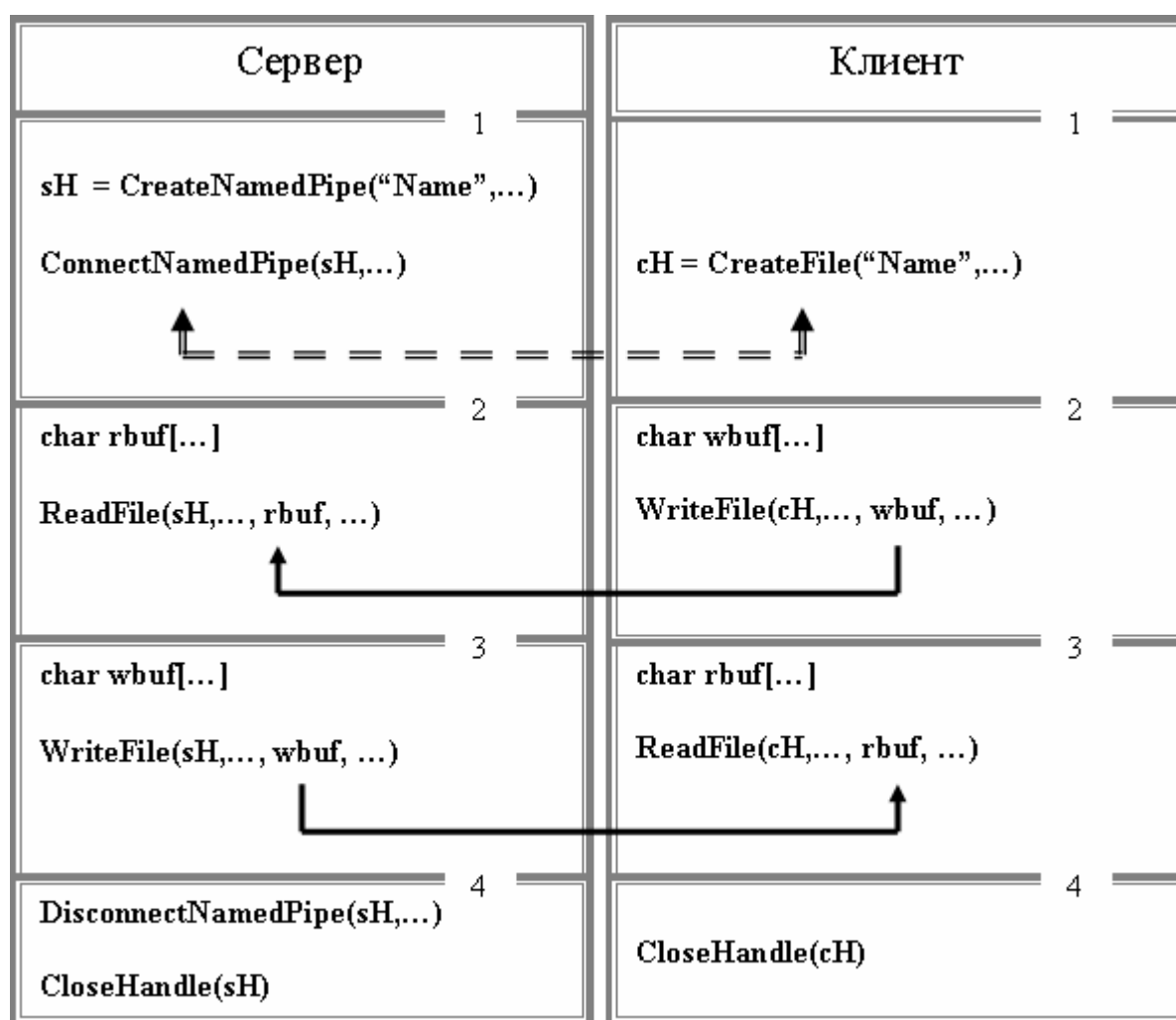


Рисунок 4.2.1. Схема взаимодействия процессов, использующих Named Pipe API

Следует сразу отметить, что при создании именованного канала в программе на языке C++, одновременно создается дескриптор (HANDLE), который потом используется другими функциями Named Pipe API для работы с данным экземпляром именованного канала. По окончании работы с каналом необходимо эти дескрипторы закрыть с помощью функции CloseHandle Win32 API.

На рисунке 4.2.1 изображены две программы, реализующие два процесса распределенного приложения. Каждая из программ разбита на четыре блока. Сплошными направленными линиями обозначается движение данных от одного процесса к другому. Прерывистой линией обозначается синхронизация процессов.

В первом блоке программы сервера выполняются две функции: CreateNamedPipe (создать именованный канал) и ConnectNamedPipe (подсоединить сервер к каналу). Одним из параметров функции CreateNamedPipe является имя канала (строка), а результатом ее работы (возвращаемым значением) является дескриптор (HANDLE) канала. Функция ConnectNamedPipe приостанавливает выполнение программы клиента до момента, пока программа клиента не выполнит функцию CreateFile.

Во втором и третьем блоках программы сервера осуществляется ввод и вывод данных (функции ReadFile и WriteFile) в именованном канале. Следует обратить внимание, что функции осуществляющие ввод и вывод используют в качестве одного из своих параметров дескриптор именованного канала.

В последнем четвертом блоке программы сервер разрывает соединение с помощью функции DisconnectNamedPipe и закрывает дескриптор именованного канала.

Для программы клиента остается пояснить только первый блок, т.к. всем остальным блокам, есть аналогичные в программе сервера. В первом блоке программы клиента выполняется функция CreateFile, одним из параметров которой является строка с именем канала. Если к моменту выполнения функции канал уже создан и сервер подсоединился к каналу, то функция CreateFile возвращает дескриптор именованного канала, который потом используется в других функциях программы клиента. Иногда перед выполнением функции CreateFile, выполняют функцию WaitNamedPipe, позволяющую определить доступность экземпляра канала. Назначение всех функций будет пояснено ниже.

Как уже отмечалось, передача данных может осуществляться как потоком, так и сообщениями. Передача данных потоком возможна в том случае, если сервер и клиент работают на одном компьютере и использует локальные имена канала, в других случаях передача данных осуществляется сообщениями. Схема изображенная на рисунке 4.2.1 является общей для этих двух случаев.

В случае функции Named Pipe API завершения с ошибкой, все функции формируют код системной ошибки Windows, который может быть получен с помощью функции GetLastError.

4.3. Создание именованного канала

Сервером именованного канала является процесс, создающий именованный канал. Именованный канал создается с помощью функции `CreateNamedPipe` (рисунок 4.3.1).

```
// -- создать именованный канал
// Назначение: функция предназначена для создания
//              именованного канала

HANDLE CreateNamedPipe
(
    LPCTSTR      pname, // [in] символическое имя канала
    DWORD        omode, // [in] атрибуты канала
    DWORD        pmode, // [in] режимы передачи данных
    DWORD        pimax, // [in] макс. к-во экземпляров канала
    DWORD        osize, // [in] размер выходного буфера
    DWORD        isize, // [in] размер входного буфера
    DWORD        timeo, // [in] время ожидания связи с клиентом
    LPSECURITY_ATTRIBUTES sattr // [in] атрибуты безопасности
);

// Код возврата: в случае успешного завершения функция
//              возвращает дескриптор именованного канала, иначе
//              возможны следующие значения:
//              INVALID_HANDLE_VALUE - неудачное завершение;
//              ERROR_INVALID_PARAMETER - значение параметра pimax
//              превосходит величину PIPE_UNLIMITED_INSTANCES
// Примечание: pname - указывает на строку именем канала в
//              локальном формате;
//              omode - задает флаги направления передачи, например
//              флаг FILE_ACCESS_DUPLEX разрешает чтение и запись в
//              канал; помимо направления здесь могут быть заданы
//              флаги асинхронной передачи, режимы буферизации и
//              безопасности;
//              pmode - задает флаги способов передачи данных,
//              например, флаг PIPE_TYPE_MESSAGE|PIPE_WAIT разрешает
//              запись данных сообщениями в синхронном режиме, а флаг
//              PIPE_READTYPE_MESSAGE|PIPE_WAIT разрешает чтение
//              сообщений в синхронном режиме;
//              pimax - максимальное количество экземпляров канала,
//              значение должно находиться в пределах от 1 до
//              PIPE_UNLIMITED_INSTANCES;
//              osize, isize - значения рассматриваются Windows
//              только как пожелания пользователя (рекомендуется 0);
//              timeo - параметр устанавливает время ожидания связи с
//              сервером в миллисекундах для функции WaitNamedPipe
//              с параметром NMWAIT_USE_DEFAULT_WAIT; может быть
//              установлено значение INFINITE (ждать бесконечно);
//              sattr - для установки атрибутов безопасности
//              по умолчанию, следует установить значение NULL
```

Рисунок 4.3.1. Функция `CreateNamedPipe`

Первый параметр функции `CreateNamedPipe` – указатель на строку имени канала. В зависимости от контекста в функциях используется два формата имени канала: локальный формат (рисунок 4.3.2) и сетевой формат (рисунок 4.3.3).

```
\\.\pipe\xxxxx
```

где: **точка (.)** – обозначает локальный компьютер;
pipe – фиксированное слово;
xxxxx – имя канала

Рисунок 4.3.2. Локальный формат имени канала

```
\\servername\pipe\xxxxx
```

где: **servername** – имя компьютера – сервера именованного канала;
pipe – фиксированное слово;
xxxxx – имя канала

Рисунок 4.3.3. Сетевой формат имени канала

Для связи сервера по одному именованному каналу с несколькими клиентами, сервер должен создать несколько экземпляров этого канала. Каждый экземпляр создается функцией `CreateNamedPipe`, которая возвращает дескриптор экземпляра именованного канала. Отметим, что поток создающий экземпляр именованного канала должен иметь право доступа `FILE_CREATE_PIPE_INSTANCE`. Этим правом по умолчанию обладает поток создавший канал. Подробнее о применении нескольких экземпляров одного канала можно ознакомиться в [4].

```
// -- соединить сервер с именованным каналом
// Назначение: функция предназначена для ожидания сервером
//               подсоединения к экземпляру именованного канала
//               клиента

BOOL ConnectNamedPipe
(
    HANDLE hP, // [in] дескриптор именованного канала
    LPOVERLAPPED ol // [in,out] используется для асинхр. связи
);

// Код возврата: в случае успешного завершения функция
//               возвращает TRUE, иначе FALSE
// Примечание: параметр ol используется только в том случае,
//               если используется асинхронная связь, в случае
//               синхронной связи можно установить значение NULL
```

Рисунок 4.3.4. Функция `ConnectNamedPipe`

После того, как сервер создал именованный канал, он должен дожидаться соединения клиента с этим сервером. Для этого сервер должен вызвать функцию `ConnectNamedPipe` (рисунок 4.3.4).

На рисунке 4.3.5 изображен фрагмент программы сервера. С помощью функции `CreateNamedPipe` создается дуплексный, синхронный канал с именем `ConsolePipe`, предназначенный для передачи сообщений. Созданный канал может иметь только один экземпляр и, если программа клиента будет использовать для проверки доступности канала `ConsolePipe` функцию `WaitNamedPipe` с параметром `NMPWAIT_USE_DEFAULT_WAIT`, то будет установлен бесконечный интервал ожидания.

```
//.....  
HANDLE hPipe; // дескриптор канала  
try  
{  
    if ((hPipe = CreateNamedPipe("\\\\.\\pipe\\ConsolePipe",  
                                PIPE_ACCESS_DUPLEX,           // дуплексный канал  
                                PIPE_TYPE_MESSAGE|PIPE_WAIT,   // сообщения | синхронный  
                                1, NULL, NULL,                 // максимум 1 экземпляр  
                                INFINITE, NULL)) == INVALID_HANDLE_VALUE)  
        throw SetPipeError("create:", GetLastError());  
    if (!ConnectNamedPipe(hPipe, NULL))                       // ожидать клиента  
        throw SetPipeError("connect:", GetLastError());  
    //.....  
}  
catch (string ErrorPipeText)  
{cout << endl << ErrorPipeText;}  
//.....
```

Рисунок 4.3.5. Фрагмент программы сервера

В программе, приведенной на рисунке 4.3.5, для обработки ошибок используется функция `SetPipeError`. Эта функция аналогична функции `SetErrorMsgText`, которая использовалась в примерах третьей главы для обработки ошибок Winsock2 API.

4.4. Соединение клиентов с именованным каналом

Прежде чем соединиться с именованным каналом, клиент может определить: доступен ли какой либо экземпляр этого канала. С этой целью клиент может вызывать функцию `WaitNamedPipe` (рисунок 4.3.1).

После обнаружения свободного канала, клиент может установить связь с каналом помощью функции `CreateFile` (рисунок 4.3.2). После успешного выполнения функции клиент и сервер могут обмениваться данными.

Здесь не рассматриваются атрибуты безопасности, которые могут быть определены при вызовах функций `CreateNamedPipe` и `CreateFile`. Однако, поясним, что для организации обмена, необходимо, чтобы атрибуты безопасности в этих функция были согласованными.

```

// -- определить доступность канала
// Назначение: функция предназначена для ожидания клиентом
//              доступного именованного канала

BOOL WaitNamedPipe
(
    LPCTSTR    pn, // [in] символическое имя канала
    DWORD      to  // [in] интервал ожидания (мс)
);

// Код возврата: в случае успешного завершения функция
//              возвращает TRUE, иначе FALSE
// Примечание: - если используется локальный канал, то имя
//              канала задается в локальном формате, если же канал
//              создан на другом компьютере, то имя канала следует
//              задавать в сетевом формате;
//              - параметр to определяет интервал времени
//              ожидания (в миллисекундах) освобождения экземпляра
//              канала; если для параметра to установлено значение
//              NMWAIT_USE_DEFAULT_WAIT, то интервал определяется
//              параметром timeo функции CreateNamedPipe; если
//              установлено значение NMWAIT_WAIT_FOREVER, то время
//              ожидания бесконечно

```

Рисунок 4.4.1. Функция WaitNamedPipe

При установке в функциях Named Pipe API атрибутов безопасности по умолчанию, как это сделано во всех приведенных здесь примерах, подсоединиться каналу удаленный клиент сможет только в том случае, если он запущен от того же имени пользователя и с тем же паролем, что и сервер. С применением атрибутов безопасности в Named Pipe API можно ознакомиться в [4].

Кроме того, следует обратить внимание на правильное использование имени канала. На рисунке 4.3.5 при создании канала с помощью функции CreateNamedPipe использовалось имя канала [\\.\pipe\ConsolePipe](#). При записи строки с именем канала в программе на языке C++, символ обратного слеша, в соответствии с правилами языка, удваиваются.

При использовании форматов имени канала, необходимо помнить, что:

- 1) при создании канала всегда используется локальный формат имени;
- 2) если клиент удаленный (на другом компьютере), то он всегда должен использовать сетевой формат имени; при этом обмен данными между клиентом и сервером осуществляется сообщениями;
- 3) если клиент локальный и использует сетевой формат имени при подсоединении к каналу (функция CreateFile), то обмен данными осуществляется сообщениями;
- 4) если клиент локальный и использует локальный формат имени канала, то обмен данными осуществляется потоком.

```

// -- открыть канал
// Назначение: функция предназначена для подключения клиента
// к именованному каналу

HANDLE CreateFile
(
    LPCTSTR    pname, // [in] символическое имя канала
    DWORD      accss, // [in] чтение или запись в канал
    DWORD      share, // [in] режим совместного использования
    LPSECURITY_ATTRIBUTES sattr // [in] атрибуты безопасности
    DWORD      oflag, // [in] флаг открытия канала
    DWORD      aflag, // [in] флаги и атрибуты
    HANDLE      exten, // [in] дополнительные атрибуты
);

// Код возврата: в случае успешного завершения функция
// возвращает дескриптор именованного канала, иначе
// INVALID_HANDLE_VALUE - неудачное завершение
// Примечание: - параметр pname указывается в локальном или
// сетевом формате: в зависимости от способа применения
// - параметр accss может принимать значения GENERIC_READ
// (чтение), GENERIC_WRITE (запись) или
// GENERIC_READ | GENERIC_WRITE (запись, чтение);
// - параметр share может принимать значения
// FILE_SHARE_READ (совместное чтение),
// FILE_SHARE_WRITE (совместная запись),
// FILE_SHARE_READ | FILE_SHARE_WRITE (чтение и запись);
// - параметр sattr для установки атрибутов безопасности
// по умолчанию, следует установить значение NULL;
// - значение параметра oflag всегда устанавливается в
// OPEN_EXISTING (открытие существующего канала);
// - значение параметров aflag и exten можно установить в
// NULL, что соответствует значениям по умолчанию

```

Рисунок 4.4.2. Функция CreateFile

Если не существует экземпляров именованного канала с тем именем, которое указано в параметре функции WaitNamedPipe, то эта функция немедленно заканчивается неудачей (FALSE), независимо от установленного в параметре функции значения интервала ожидания. Если же канал создан, но сервер не выполнил функцию ConnectNamedPipe, то функция WaitNamedPipe на стороне клиента все равно вернет FALSE и сформирует диагностический код (функции GetLastError) ERROR_PIPE_CONNECTED. Даже в том случае, если функция WaitNamedPipe обнаружит свободный экземпляр канала (и вернет TRUE), то все равно нет гарантии, что до выполнения функции CreateFile этот канал не будет занят другим клиентом. Все эти замечания, делают применение функции WaitNamedPipe в большинстве случаев нецелесообразным. Фрагмент программы клиента,

демонстрирующий подключение к именованному каналу изображен на рисунке 4.4.3.

```
//.....
HANDLE hPipe; // дескриптор канала
try
{
if ((hPipe = CreateFile(
    "\\.\pipe\ConsolePipe",
    GENERIC_READ|GENERIC_WRITE,
    FILE_SHARE_READ|FILE_SHARE_WRITE,
    NULL, OPEN_EXISTING, NULL,
    NULL)) == INVALID_HANDLE_VALUE)
    throw SetPipeError("createfile:", GetLastError());
//.....
}
catch (string ErrorPipeText)
{cout << endl << ErrorPipeText;}
//.....
```

Рисунок 4.4.3. Фрагмент программы клиента

4.5. Обмен данными по именованному каналу

Для обмена данными по именованному каналу используются три функции: ReadFile, WriteFile и PeekNamedPipe (рисунки 4.5.1, 4.5.2, 4.5.3).

```
// -- читать данные из канала
// Назначение: функция предназначена чтения данных из
//              именованного канала

BOOL ReadFile
(
    HANDLE      hP,    // [in] дескриптор канала
    LPVOID      pb,    // [out] указатель на буфер ввода
    DWORD       sb,    // [in] количество читаемых байт
    LPDWORD     ps,    // [out] количество прочитанных байт
    LPOVERLAPPED ol    // [in,out] для асинхронной обработки
);

// Код возврата: в случае успешного завершения функция
//              возвращает TRUE, иначе FALSE
// Примечание: если не используется асинхронная обработка
//              параметр ol рекомендуется установить в NULL
```

Рисунок 4.5.1. Функция ReadFile

```

// -- писать данные в канал
// Назначение: функция предназначена записи данных в
//              именованный канал

BOOL WriteFile
(
    HANDLE      hP,    // [in] дескриптор канала
    LPVOID      pb,    // [in] указатель на буфер вывода
    DWORD       sb,    // [in] количество записываемых байт
    LPDWORD     ps,    // [out] количество записанных байт
    LPOVERLAPPED ol    // [in,out] для асинхронной обработки
);

// Код возврата: в случае успешного завершения функция
//              возвращает TRUE, иначе FALSE
// Примечание: если не используется асинхронная обработка
//              параметр ol рекомендуется установить в NULL

```

Рисунок 4.5.2. Функция WriteFile

Параметры функций ReadFile и WriteFile достаточно просты и не требуют дополнительного пояснения. Функция PeekNamedPipe копирует данные из канала в буфер. При этом данные не извлекаются и их еще можно считать (извлечь) с помощью функции ReadFile.

```

// -- копировать данные канала
// Назначение: функция предназначена для получения данных
//              из канала без извлечения

BOOL PeekNamedPipe
(
    HANDLE      hP,    // [in] дескриптор канала
    LPVOID      pb,    // [out] указатель на буфер
    DWORD       sb,    // [in] размер буфера
    LPDWORD     pi,    // [out] количество прочитанных байт
    LPDWORD     pa,    // [out] количество доступных байт
    LPDWORD     pr,    // [out] количество непрочитанных байт
);

// Код возврата: в случае успешного завершения функция
//              возвращает TRUE, иначе FALSE

```

Рисунок 4.5.3. Функция PeekNamedPipe

4.6. Передача транзакций по именованному каналу

Для обмена сообщениями по сети может использоваться функция `TransactNamedPipe` (рисунок 4.6.1), которая объединяет операции чтения и записи в одну операцию. Такую объединенную операцию называют *транзакцией* именованного канала. Функция `TransactNamedPipe` может быть использована только в том случае, если сервер именованного канала установил флаг `PIPE_TYPE_MESSAGE`.

Применение `TransactNamedPipe` целесообразно, если другая сторона канала может обеспечить достаточно быструю реакцию и оправить ответ на пришедшее сообщение.

```
// -- писать и читать данные канала
// Назначение: функция предназначена для выполнения записи в
// канал и чтения из канала за одну операцию

BOOL TransactNamedPipe
(
    HANDLE      hP,    // [in] дескриптор канала
    LPVOID      pw,    // [in] указатель на буфер для записи
    DWORD       sw,    // [in] размер буфера для записи
    LPVOID      pr,    // [out] указатель на буфер для чтения
    DWORD       sr,    // [in] размер буфера для чтения
    LPDWORD     pr,    // [out] количество прочитанных байт
    LPOVERLAPPED ol     // [in,out] для асинхронного доступа
) ;

// Код возврата: в случае успешного завершения функция
// возвращает TRUE, иначе FALSE
// Примечание: параметр ol используется для асинхронного
// доступа к каналу, если асинхронный доступ не
// предполагается, то следует указать NULL
```

Рисунок 4.6.1. Функция `TransactNamedPipe`

Часто взаимодействие сервера и клиента сводится к простому запросу клиента к серверу для получения некоторого сервиса. После выполнения запрошенной клиентом сервисной услуги, сервер информирует клиента о результате своей работы. Т.е. речь идет об одиночных эпизодических транзакциях.

Если требуется передать только одну транзакцию, то используют функцию `CallNamedPipe` (рисунок 4.6.2), которая работает следующим образом.

Сначала осуществляется установка связи с именованным каналом, имя которого указывается в параметрах функции. При этом именованный канал должен быть открыт в режиме данных сообщениями. После установки связи функция пересылает в канал единственное сообщение и получает одно

сообщение в ответ. После обмена данными осуществляется разрыв связи с именованным каналом.

```
// -- выполнить одну транзакцию
// Назначение: функция предназначена для установки связи с
//              именованным каналом, выполнения одной транзакции
//              и разрыва связи

BOOL CallNamedPipe
(
    LPCTSTR      nP,    // [in] указатель на имя канала
    LPVOID       pw,    // [in] указатель на буфер для записи
    DWORD        sw,    // [in] размер буфера для записи
    LPVOID       pr,    // [out] указатель на буфер для чтения
    DWORD        sr,    // [in] размер буфера для чтения
    LPDWORD      pr,    // [out] количество прочитанных байт
    DWORD        to     // [in] интервал ожидания
) ;

// Код возврата: в случае успешного завершения функция
//               возвращает TRUE, иначе FALSE
// Примечание: параметр to устанавливает интервал времени в
//             миллисекундах; кроме того, здесь могут быть
//             установлены те же значения, что и в функции
//             WaitNamedPipe
```

Рисунок 4.6.2. Функция CallNamedPipe

4.7. Определение состояния и изменение характеристик именованного канала

Для получения информации о созданном именованном канале можно использовать две функции `GetNamedPipeInfo` и `GetNamedPipeHandleState` (рисунки 4.7.1 и 4.7.2).

Функция `GetNamedPipeInfo` используется для получения информации об атрибутах именованного канала, который являются статическими и не могут быть изменены. Входными параметрами служат дескриптор и тип канала, о котором предполагается получить информацию. При завершении, функция возвращает значения размеров буферов ввода и вывода, а также максимальное количество экземпляров данного именованного канала.

Т.к. для получения информации используется дескриптор, то предполагается, что перед выполнением функции `GetNamedPipeInfo`, канал уже создан (функция `CreateNamedPipe` на стороне сервера) или открыт (функция `CreateFile` на стороне клиента). При этом для ее выполнения необходимо, чтобы было разрешено чтение канала.

```

// -- получить информацию об именованном канале
// Назначение: функция предназначена для получения
//              статических характеристик именованного канала
BOOL GetNamedPipeInfo
(
    HANDLE hP,      // [in] дескриптор именованного канала
    LPDWORD pfg,    // [in] указатель на флаг-тип канала
    LPDWORD psw,    // [out] указатель на размер выходного буфера
    LPDWORD psr,    // [out] указатель на размер входного буфера
    LPDWORD pmi,    // [out] указатель на макс. к-во экземпляров канала
);
// Код возврата: в случае успешного завершения функция
//              возвращает TRUE, иначе FALSE
// Примечание: параметр pfg указывает на переменную типа
//              DWORD, в которой установлен тип именованного канала,
//              атрибуты которого запрашиваются; для установки
//              этой переменной должны использоваться константы:
//              PIPE_CLIENT_END, PIPE_SERVER_END - для обозначения
//              типа используемого в функции дескриптора;
//              PIPE_TYPE_BYTE, PIPE_TYPE_MESSAGE - для установки
//              типа передачи (поток и сообщения)

```

Рисунок 4.7.1. Функция GetNamedPipeInfo

```

// -- получить состояния именованного канала
// Назначение: функция предназначена для получения
//              динамических характеристик именованного канала
BOOL GetNamedPipeHandleState
(
    HANDLE hP,      // [in] дескриптор именованного канала
    LPDWORD pst,    // [out] указатель на состояние канала
    LPDWORD pci,    // [out] указатель на к-во экземпляров каналов
    LPDWORD pcc,    // [out] указатель на макс. к-во байт
    LPDWORD pto,    // [out] указатель на интервал задержки
    LPTSTR pun,    // [out] указатель на имя владельца канала
    DWORD lun       // [in] длина буфера для имени владельца канала
);
// Код возврата: в случае успешного завершения функция
//              возвращает TRUE, иначе FALSE
// Примечание: - параметр pst указывает на переменную типа
//              DWORD, в которой установлена комбинация значений:
//              PIPE_NOWAIT - канал не блокирован;
//              PIPE_READMODE_MESSAGE - канал открыт в режиме
//              передачи сообщениями;
//              - параметр pcc - указывает на максимальное
//              количество байтов, которые клиент именного канала
//              должен записать в канал перед передачей серверу;
//              - параметр pto - указывает на количество миллисекунд
//              которые должно пройти прежде, чем данные будут
//              переданы

```

Рисунок 4.7.2. Функция GetNamedPipeHandleState

Чаще всего функция `GetNamedPipeInfo` используется на стороне клиента после открытия канала для выяснения размеров буферов, установленных операционной системой при создании канала.

Функция `GetNamedPipeHandleState` используется для получения динамических параметров (которые могут быть изменены) именованного канала. Для изменения некоторых параметров может быть использована функция `SetNamedPipeHandleState` (рисунок 4.7.3).

```
// -- изменить характеристики канала
// Назначение: функция предназначена для изменения
//              динамических характеристик именованного канала

BOOL SetNamedPipeHandleState
(
    HANDLE hP,      // [in] дескриптор именованного канала
    LPDWORD pst,    // [in] указатель на новое состояние канала
    LPDWORD pcc,    // [in] указатель на макс. к-во байтов
    LPDWORD pto     // [in] указатель на интервал задержки
);

// Код возврата: в случае успешного завершения функция
//              возвращает TRUE, иначе FALSE
```

Рисунок 4.7.2. Функция `SetNamedPipeHandleState`

Более подробно о применении функций определения и изменения состояния именованного канала описано в [4].

4.8. Итоги главы

1. Современные операционные системы имеют встроенные механизмы межпроцессорного взаимодействия (IPC), позволяющие создавать распределенные в локальной сети приложения. Для работы использования IPC-механизмов операционные системы предоставляют специальные программные интерфейсы.
2. Интерфейс Named Pipe (именованный канал) реализует один из IPC-механизмов операционной системы Windows и позволяет создавать распределенные приложения архитектуры клиент-сервер.
3. Именованный канал представляет собой объект операционной системы Windows, позволяющий создавать между распределенными в локальной TCP/IP-сети процессами дуплексные и полудуплексные каналы, по которым может осуществляться передача данных в синхронном или асинхронном режимах.
4. В состав интерфейса Named Pipe входят функции для управления каналом, функции для обмена данными по каналу и функции работы с транзакциями.