



СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

ЛЕКЦИЯ № 8

ПРЕПОДАВАТЕЛЬ: ХУСТОЧКА А.В.



СОБЫТИЯ

HANDLE

WINAPI

```
CreateEventW(  
    _In_opt_ LPSECURITY_ATTRIBUTES lpEventAttributes,  
    _In_ BOOL bManualReset,  
    _In_ BOOL bInitialState,  
    _In_opt_ LPCWSTR lpName  
);
```

HANDLE

WINAPI

```
OpenEventW(  
    _In_ DWORD dwDesiredAccess,  
    _In_ BOOL bInheritHandle,  
    _In_ LPCWSTR lpName  
);
```

BOOL

WINAPI

```
SetEvent(  
    _In_ HANDLE hEvent  
);
```

BOOL

WINAPI

```
ResetEvent(  
    _In_ HANDLE hEvent  
);
```

BOOL

WINAPI

```
PulseEvent(  
    _In_ HANDLE hEvent  
);
```

ТАЙМЕРЫ

```
HANDLE
WINAPI
CreateWaitableTimerW(
    _In_opt_ LPSECURITY_ATTRIBUTES
lpTimerAttributes,
    _In_ BOOL bManualReset,
    _In_opt_ LPCWSTR lpTimerName
);
```

```
HANDLE
WINAPI
OpenWaitableTimerW(
    _In_ DWORD dwDesiredAccess,
    _In_ BOOL bInheritHandle,
    _In_ LPCWSTR lpTimerName
);
```

```
BOOL
WINAPI
SetWaitableTimer(
    _In_ HANDLE hTimer,
    _In_ const LARGE_INTEGER* lpDueTime,
    _In_ LONG lPeriod,
    _In_opt_ PTIMERAPCROUTINE pfnCompletionRoutine,
    _In_opt_ LPVOID lpArgToCompletionRoutine,
    _In_ BOOL fResume
);
```

```
BOOL
WINAPI
CancelWaitableTimer(
    _In_ HANDLE hTimer
);
```

ОРГАНИЗАЦИЯ СВЯЗИ МЕЖДУ ПРОЦЕССАМИ

ОС создаёт некоторую передающую среду, аналог общей памяти, представляющую собой разделяемый ресурс, к которому имеют доступ оба потока/процесса. Если оформить этот ресурс как общий файл, то получим простейший канал передачи данных.

Организация связи между процессами всегда предполагает установления следующих характеристик:

- Направление связи: однонаправленная или двунаправленная
- Используемая модель передачи данных: потоковая или модель сообщений
- Объём передаваемой информации и сведения о том, обладает ли канал буффером необходимого размера
- Синхронность обмена данными: синхронный или асинхронный

ПРИНЦИП КАНАЛА

- Основной принцип работы канала состоит в буферизации вывода одного процесса и обеспечении возможности чтения содержимого программного канала другим процессом.
- При этом интерфейс программного канала совпадает с интерфейсом обычного файла и реализуется файловыми операциями read и write. Двумя основными моделями передачи данных по каналу являются поток ввода-вывода и сообщения. При передаче в рамках потоковой модели данные представляют собой неструктурированную последовательность байтов и никак не интерпретируются системой. В модели сообщений на передаваемые данные накладывается некоторая структура, обычно их разделяют на сообщения заранее оговоренного формата.

МЕХАНИЗМА ГЕНЕРАЦИИ КАНАЛА

- Механизм генерации канала предполагает получение процессом-создателем (процессом-сервером) двух дескрипторов для пользования этим каналом. Один из дескрипторов применяется для чтения из канала, другой - для записи в канал. Для общения с каналом процесс-клиент и процесс-сервер должны иметь дескрипторы для чтения и записи.
- Процесс-сервер получает дескриптор при создании канала. Процесс-клиент может получить дескриптор в результате наследования, в том случае, когда клиент является потомком сервера. Это типично для общения через так называемые анонимные каналы, которые функционируют в пределах локальной вычислительной системы, обеспечивают передачу данных между процессами, выполняющимися на одном компьютере, и хорошо подходят для перенаправления выходного потока одной программы на вход другой.
- Другой способ получения дескриптора - открытие по имени уже существующего именованного канала неродственным процессом, который в результате также становится обладателем необходимых описателей. Такой способ используется в сетевых взаимодействиях.

ИМЕНОВАННЫЕ КАНАЛЫ

- Именованным каналом называется объект ядра операционной системы, который обеспечивает передачу данных между процессами, выполняющимися на компьютерах в одной локальной сети.
- Именованные каналы должны иметь уникальные в рамках сети имена в соответствии с правилами именования ресурсов в сетях Windows (Universal Naming Convention, UNC):

\\ServerName\pipe\PipeName

- Для общения внутри одного компьютера имя записывается в форме `\\.\pipe\PipeName`, где `ServerName` – сетевое имя компьютера, на котором выполняется процесс-сервер; вместо сетевого имени может также использоваться его IP-адрес. "." в этом поле вместо `ServerName` обозначает, что процесс-сервер, к которому производится обращение, находится на данной локальной машине. Слово "pipe" в составе имени фиксировано, а `PipeName` – имя канала, задаваемое пользователем.

ИМЕНОВАННЫЕ КАНАЛЫ

- Если клиент и сервер именованного канала работают на одном компьютере, то для связи клиента с именованным каналом нужно вводить в качестве имени сервера или точку, или полное имя компьютера.
- Если вместо имени компьютера будет введена точка, то именованный канал откроется в режиме передачи данных потоком, а не сообщениями.
- Поскольку имена каналов интегрированы в общую структуру имен объектов, а их интерфейс совпадает с интерфейсом файлов, приложения могут открывать именованные каналы с помощью функции `CreateFile` и взаимодействовать с ними через функции `ReadFile` и `WriteFile`.

ИМЕНОВАННЫЕ КАНАЛЫ

Порядок работы с именованными каналами:

- создание именованного канала сервером
- соединение сервера с экземпляром именованного канала
- соединение клиента с экземпляром именованного канала
- обмен данными по именованному каналу
- отсоединение сервера от экземпляра именованного канала
- закрытие именованного канала клиентом и сервером.

ИМЕНОВАННЫЕ КАНАЛЫ

HANDLE

WINAPI

```
CreateNamedPipeW(  
_In_ LPCWSTR lpName,           // имя канала  
_In_ DWORD dwOpenMode,         // атрибуты  
_In_ DWORD dwPipeMode,         // режим передачи данных  
_In_ DWORD nMaxInstances,       // максимальное количество экземпляров канала  
_In_ DWORD nOutBufferSize,      // размер выходного буфера  
_In_ DWORD nInBufferSize,       // размер входного буфера  
_In_ DWORD nDefaultTimeOut,     // время ожидания связи с клиентом  
_In_opt_ LPSECURITY_ATTRIBUTES lpSecurityAttributes // атрибуты безопасности  
);
```

СОЗДАНИЕ ИМЕНОВАННОГО КАНАЛА

Параметр lpName указывает на строку, которая должна иметь вид: `\\.\pipe\pipe_name`

Здесь "." (точка) обозначает локальную машину, т. к. новый именованный канал всегда создается на локальной машине; слово `pipe` — фиксировано, а `pipe_name` обозначает имя канала, которое задается пользователем и нечувствительно к регистру.

Параметр dwOpenMode задает флаги, которые определяют направление передачи данных, буферизацию, синхронизацию обмена данными и права доступа к именованному каналу. Для определения направления передачи данных используются флаги:

- `PIPE_ACCESS_DUPLEX` — чтение и запись в канал;
- `PIPE_ACCESS_INBOUND` — клиент пишет, а сервер читает данные
- `PIPE_ACCESS_OUTBOUND` — сервер пишет, а клиент читает данные.

Флаг, определяющий направление передачи данных по именованному каналу, должен совпадать для всех экземпляров одного и того же именованного канала. Для определения способа буферизации и синхронизации используются флаги:

- `FILE_FLAG_WRITE_THROUGH` — запрещает буферизацию при передаче данных по сети;
- `FILE_FLAG_OVERLAPPED` — разрешает асинхронную передачу данных по каналу. Эти флаги могут быть разными для каждого экземпляра одного и того же именованного канала. В этом же параметре могут быть заданы другие режимы доступа к именованному каналу.

Флаги, используемые для определения атрибутов безопасности, `PIPE_ACCESS_DUPLEX`, `PIPE_ACCESS_INBOUND` и `PIPE_ACCESS_OUTBOUND`, определяют специфические права доступа к именованному каналу, которые включают следующие родовые права доступа:

- `PIPE_ACCESS_DUPLEX` — включает родовые права доступа `GENERIC_READ`, `GENERIC_WRITE` и `SYNCHRONIZE`;
- `PIPE_ACCESS_INBOUND` — включает родовые права доступа `GENERIC_READ` и `SYNCHRONIZE`;
- `PIPE_ACCESS_OUTBOUND` — включает родовые права доступа `GENERIC_WRITE` и `SYNCHRONIZE`.

СОЗДАНИЕ ИМЕНОВАННОГО КАНАЛА

Параметр `dwPipeMode` задает флаги, определяющие способ передачи данных по именованному каналу. Для определения способов чтения и записи данных в именованный канал используются флаги:

- `PIPE_TYPE_BYTE` — запись данных потоком;
- `PIPE_TYPE_MESSAGE` — запись данных сообщениями;
- `PIPE_READMODE_BYTE` — чтение данных потоком;
- `PIPE_READMODE_MESSAGE` — чтение данных сообщениями.

По умолчанию данные по именованному каналу передаются потоком. Флаги, определяющие способ чтения и записи данных в именованный канал, должны совпадать для всех экземпляров одного и того же канала. Для определения синхронизации доступа к именованному каналу используются флаги:

- `PIPE_WAIT` — синхронная связь с каналом и обмен данными по каналу;
- `PIPE_NOWAIT` — асинхронная связь с каналом и обмен данными по каналу. Эти флаги могут быть разными для каждого экземпляра именованного канала.

СОЗДАНИЕ ИМЕНОВАННОГО КАНАЛА

- **Параметр `nMaxinstances`** определяет максимальное число экземпляров именованного канала, которое может находиться в пределах от 1 до `PIPE_UNLIMITED__INSTANCES`. Каждый экземпляр канала предназначен обмена данными по каналу между сервером и отдельным клиентом.
- **Параметры `nOutBuffersize` и `nInBuffersize`** определяют соответственно размеры выходного и входного буферов для обмена данными по именованному каналу. Однако эти значения рассматриваются операционными системами Windows только как пожелания пользователя, а сам выбор размеров буферов остается за операционной системой.
- **Параметр `nDefaultTimeOut`** устанавливает время ожидания клиентом связи с сервером. Это время используется при вызове клиентом функции `WaitNamedPipe`, в которой параметр `nTimeout` имеет значение `NMPWAIT_USE_DEFAULT_WAIT`.

СОЗДАНИЕ ИМЕНОВАННОГО КАНАЛА

- Для связи сервера с несколькими клиентами по одному именованному каналу сервер должен создать несколько экземпляров этого канала. Каждый экземпляр именованного канала создается вызовом функции `CreateNamedPipe`, которая возвращает дескриптор экземпляра именованного канала.
- Отметим, что в этом случае поток, создающий экземпляр именованного канала, должен иметь право доступа `FILE_CREATE_PIPE_INSTANCE` к именованному каналу. Этим правом по умолчанию обладает владелец именованного канала, т. е. тот процесс, который создал этот именованный канал.

СОЕДИНЕНИЕ СЕРВЕРА С КАНАЛОМ

- После того как сервер создал именованный канал, он должен дожидаться соединения клиента с этим каналом. Для этого сервер вызывает функцию `ConnectNamedPipe`, которая имеет следующий прототип:

```
BOOL
WINAPI
ConnectNamedPipe(
    _In_ HANDLE hNamedPipe,    // дескриптор канала
    _Inout_opt_ LPOVERLAPPED lpOverlapped // асинхронная связь
);
```

- В случае успешного завершения эта функция возвращает ненулевое значение, а в случае неудачи — значение `FALSE`. Сервер использует эту функцию для связи с клиентом по каждому свободному экземпляру именованного канала.

ОТСОЕДИНЕНИЕ ОТ КАНАЛА

- После окончания обмена данными с клиентом сервер может вызвать функцию DisconnectNamedPipe, которая имеет следующий прототип:

```
BOOL
WINAPI
DisconnectNamedPipe(
    _In_ HANDLE hNamedPipe // дескриптор канала
);
```

- Возвращает ненулевое значение в случае успеха или значение FALSE — в случае неудачи. Эта функция разрывает связь сервера с клиентом. После этого клиент не может обмениваться данными с сервером по данному именованному каналу, и поэтому любая операция доступа к именованному каналу со стороны клиента вызовет ошибку. После разрыва связи с одним клиентом сервер снова может вызвать функцию ConnectNamedPipe, чтобы установить связь по этому же именованному каналу с другим клиентом.

СОЕДИНЕНИЕ КЛИЕНТОВ С ИМЕНОВАННЫМ КАНАЛОМ

- Прежде чем соединяться с именованным каналом, клиент может попытаться определить, доступен ли какой-либо экземпляр этого канала для соединения. С этой целью клиент должен вызвать функцию `WaitNamedPipe`, которая имеет следующий прототип: `BOOL`

`WINAPI`

```
WaitNamedPipeW(  
    _In_ LPCWSTR lpNamedPipeName, // указатель на имя канала  
    _In_ DWORD nTimeout           // интервал ожидания  
);
```

- В случае успешного завершения эта функция возвращает ненулевое значение, а в случае неудачи — `FALSE`. Опишем параметры этой функции.
- **Параметр `lpNamedPipeName`** указывает на строку, которая должна иметь вид: `\\server_name\pipe\pipe_name`
- Здесь `server_name` обозначает имя компьютера, на котором выполняется сервер именованного канала, слово `pipe` фиксировано, а `pipe_name` задает имя именованного канала.
- **Параметр `nTimeout`** задает временной интервал, в течение которого клиент ждет связь с сервером. Этот временной интервал определяется в миллисекундах или может быть равен одному из следующих значений: - `NMPWAIT_USE_DEFAULT_WAIT` — интервал времени ожидания определяется значением параметра `nDefaultTimeout`, который задается в функции `CreateNamedPipe`; - `NMPWAIT_WAIT_FOREVER` — бесконечное время ожидания связи с именованным каналом.



Сделаем замечания относительно работы функции `WaitNamedPipe`.

- Если заранее известно, что сервер вызвал функцию `ConnectNamedPipe`, то функция соединения клиента с каналом `CreateFile` (см. далее) может вызываться без предварительного вызова функции `WaitNamedPipe`.
- Если не существует экземпляров именованного канала с именем `IpNamedPipe`, то функция `WaitNamedPipe` немедленно завершается неудачей независимо от времени ожидания, заданного параметром `nTimeOut`.
- Если клиент соединяется с каналом до вызова сервером `ConnectNamedPipe`, то функция `WaitNamedPipe` возвращает значение `FALSE` и функция `GetLastError` вернет код `ERROR_PIPE_CONNECTED`. Поэтому `WaitNamedPipe` нужно вызывать только после соединения сервера с каналом посредством функции `ConnectNamedPipe`

УСТАНОВЛЕНИЕ СВЯЗИ С КАНАЛОМ

- После того, как обнаружен свободный экземпляр канала, чтобы установить связь с этим каналом, клиент должен вызвать функцию `CreateFile`, которая имеет следующий прототип:

```
HANDLE
WINAPI
CreateFileW(
    _In_ LPCWSTR lpFileName,      // указатель на имя канала
    _In_ DWORD dwDesiredAccess,   // чтение или запись в канал
    _In_ DWORD dwShareMode,       // режим совместного использования
    _In_opt_ LPSECURITY_ATTRIBUTES lpSecurityAttributes, // атрибуты безопасности
    _In_ DWORD dwCreationDisposition, // флаг открытия канала
    _In_ DWORD dwFlagsAndAttributes, // флаги и атрибуты
    _In_opt_ HANDLE hTemplateFile  // дополнительные атрибуты
);
```

- В случае успешного завершения эта функция возвращает дескриптор имени ванного канала, а в случае неудачи — значение `INVALID_HANDLE_VALUE`.

УСТАНОВЛЕНИЕ СВЯЗИ С КАНАЛОМ

- Здесь функция используется для открытия именованного канала, и ее параметры могут принимать следующие значения:
- **Параметр lpFileName** должен указывать на имя канала, которое быть задано в том же формате, что и в функции `WaitNamedPipe`. Отметим, что если клиент работает на той же машине, что и сервер, и использует для открытия именованного канала в функции `CreateFile` имя канала как `\\.\pipe\pipe_name`, то файловая система именованных каналов открывает этот именованный канал в режиме передачи данных потоком. Чтобы открыть именованный канал в режиме передачи данных сообщениями, нужно задавать имя канала в виде:
`\\server_name\pipe\pipe_name`
- **Параметр dwDesiredAccess** может принимать одно из следующих значений:
 - 0 - разрешает получить атрибуты канала;
 - `GENERIC_READ` — разрешает чтение из канала;
 - `GENERIC_WRITE` — разрешает запись в канал.
- Следует отметить, что функция `CreateFile` завершается неудачей, если доступ к именованному каналу, заданный этими значениями, не соответствует значениям параметра `dwOpenMode` в функции `CreateNamedPipe`.

УСТАНОВЛЕНИЕ СВЯЗИ С КАНАЛОМ

- **Параметр `dwShareMode`** определяет режим совместного использования именованного канала и может принимать значение 0, которое запрещает совместное использование именованного канала, или любую комбинацию следующих значений:
 - `FILE_SHARE_READ` — разрешает совместное чтение из канала;
 - `FILE_SHARE_WRITE` — разрешает совместную запись в канал.
- **Параметр `lpSecurityAttributes`** задает атрибуты безопасности именованного канала. Пока значение этого параметра будем устанавливать в `NULL`. Для именованного канала параметр `dwCreationDisposition` должен быть равен значению `OPEN_EXISTING`, т. к. клиент всегда открывает существующий именованный канал. Для именованного канала параметр `dwFlagsAndAttributes` можно задать равным 0, что определяет флаги и атрибуты по умолчанию. Значение параметра `hTemplateFile` задается равным `NULL`.
- Несмотря на то, что функция `WaitNamedPipe` может успешно завершиться, последующий вызов функции `CreateFile` может завершиться неудачей по следующим причинам: между вызовами этих функций сервер закрыл канал или между вызовами функций другой клиент связался с экземпляром этого канала. Для предотвращения последней ситуации сервер должен создавать новый экземпляр именованного канала после каждого успешного завершения функции `ConnectNamedPipe` или создать сразу несколько экземпляров именованного канала.

ОБМЕН ДАННЫМИ МЕЖДУ КАНАЛАМИ

- Для обмена данными по именованному каналу используются функции `ReadFile` и `WriteFile`. Так как в случае именованного канала разрешен асинхронный обмен данными, то в функциях `ReadFile` и `WriteFile` может использоваться параметр `lpOverlapped` при том условии, что в вызове функции `CreateNamedPipe` в параметре `dwOpenMode` был установлен флаг `FILE_FLAG_OVERLAPPED`.
- Максимально в именованный канал может быть записано до 65 535 байт одной операцией `WriteFile`. Кроме того, отметим, что для асинхронного обмена данными по именованному каналу могут использоваться также функции `ReadFileEx` и `WriteFileEx`.
- После завершения обмена данными по именованному каналу потоки должны закрыть дескрипторы экземпляров именованного канала, используя функцию `CloseHandle`.

ФУНКЦИИ WINAPI ДЛЯ РАБОТЫ С ИМЕНОВАННЫМИ КАНАЛАМИ

Функция	Описание
CreateNamedPipe	Создание именованного канала или нового экземпляра канала. Функция доступна только серверу.
CreateFile	Подключение к экземпляру именованного канала со стороны клиента. Функция доступна только клиенту.
WaitNamedPipe	Ожидание клиентом появления свободного экземпляра именованного канала для подключения к нему.
ConnectNamedPipe	Ожидание сервером подключения клиента к экземпляру именованного канала.
ReadFile	Чтение данных из именованного канала. Функция доступна как клиенту, так и серверу.
WriteFile, WriteFileEx	Запись данных в именованный канал. Функция доступна как клиенту, так и серверу.

ФУНКЦИИ WINAPI ДЛЯ РАБОТЫ С ИМЕНОВАННЫМИ КАНАЛАМИ

Функция	Описание
PeekNamedPipe	Чтение данных из именованного канала без удаления прочитанных данных из буфера канала. Функция доступна как клиенту, так и серверу.
TransactNamedPipe	Запись и чтение из именованного канала одной операцией. Функция доступна как клиенту, так и серверу.
DisconnectNamedPipe	Отсоединение сервера от экземпляра именованного канала.
WriteFile, WriteFileEx	Запись данных в именованный канал. Функция доступна как клиенту, так и серверу.
GetNamedPipeInfo	Получение информации об именованном канале.
GetNamedPipeHandleState	Получение текущего режима работы именованного канала и количества созданных экземпляров канала.
SetNamedPipeHandleState	Установка текущего режима работы именованного канала.
CloseHandle	Закрытие дескриптора экземпляра именованного канала, освобождение связанных с объектом ресурсов

КОПИРОВАНИЕ ДАННЫХ ИЗ ИМЕНОВАННОГО КАНАЛА

- Для копирования данных из именованного канала используется функция PeekNamedPipe, которая копирует данные в буфер, не удаляя их из канала. Эта функция имеет следующий прототип:

BOOL

WINAPI

```
PeekNamedPipe(  
    _In_ HANDLE hNamedPipe,      // дескриптор канала  
    _Out_writes_bytes_to_opt_(nBufferSize, *lpBytesRead) LPVOID lpBuffer, // буфер данных  
    _In_ DWORD nBufferSize,      // размер буфера данных  
    _Out_opt_ LPDWORD lpBytesRead, // количество прочитанных байтов  
    _Out_opt_ LPDWORD lpTotalBytesAvail, // количество доступных байтов  
    _Out_opt_ LPDWORD lpBytesLeftThisMessage // количество непрочитанных байтов  
);
```

- В случае успешного завершения функция возвращает ненулевое значение, а в случае неудачи – FALSE. Если данных в канале нет, то функция немедленно возвращает управление и устанавливает в ноль значения переменных, на которые указывают параметры lpBytesRead, lpTotalBytesAvail и lpBytesLeftThisMessage.

КОПИРОВАНИЕ ДАННЫХ ИЗ ИМЕНОВАННОГО КАНАЛА

Параметры функции PeekNamedPipe

- дескриптор именованного канала (должен быть открыт в режиме чтения);
- область памяти, куда будут читаться данные из канала (если установлены в NULL, то данные читаться не будут);
- размер этой области памяти; количество прочитанных байт (если данные не читаются, то - NULL);
- количество доступных байт в канале; количество непрочитанных байт из сообщения.

Если данные передаются потоком, то функция читает ровно столько байтов данных, какова длина буфера данных. Если же данные передаются сообщениями, то функция читает полностью сообщение, которое входит в буфер данных. В противном случае читается часть сообщения, а количество непрочитанных байтов возвращается через переменную, на которую указывает параметр lpBytesLeftThisMessage.