**Почтовые каналы передачи данных Mailslot**

Способы организации передачи данных

Рассмотрим простой способ организации передачи данных между различными процессами, основанный на использовании дейтаграммных каналов Mailslot.

Каналы Mailslot позволяют выполнять одностороннюю передачу данных от одного или нескольких клиентов к одному или нескольким серверам.

Главная особенность каналов Mailslot заключается в том, что они, в отличие от других средств, рассмотренных нами, позволяют передавать данные в широковещательном режиме.

Это означает, что на компьютере или в сети могут работать несколько серверных процессов, способных получать сообщения через каналы Mailslot.

При этом один клиентский процесс может посылать сообщения сразу всем этим серверным процессам.

С помощью каналов Pipe вы не сможете передавать данные в широковещательном режиме, так как только два процесса могут создать канал типа Pipe.

MailSlot - это файл, находящийся в памяти, доступ к которому осуществляется стандартными файловыми функциями Win32. Данные в Mailslot могут быть в любой форме, но общий размер не может быть больше 64K.

Различают два типа процессов: MailSlot-сервер и MailSlot-клиент.

MailSlot-cервер - является процессом, который создает и обладает MailSlot. Когда сервер создает MailSlot, он получает хэндл.

Этот хэндл должен использоваться, когда процесс читает сообщения от MailSlot. Только процесс, который создает MailSlot или получает хэндл некоторым другим механизмом, может прочитать данные из MailSlot. Все MailSlot локальные на процессе, который создает их.

Процесс не может создать дистанционный MailSlot.

MailSlot-клиент - является процессом, который пишет сообщение в MailSlot. Любой процесс, который имеет имя MailSlot может записать в него информацию

Таким образом, MailSlot представляет собой простой и однонаправленный интерфейс взаимодействия между процессами. Процесс-сервер может завести почтовый ящик и дать ему имя, глобальное в сети. Любой клиент может с помощью операций работы с файлами отправить данные в этот ящик. Сервер, по мере необходимости, может читать переданные ему данные. Кроме этого, возможно широковещательная передача информации клиентом всем серверам домена.

Рассмотрим наиболее важные функции программного интерфейса Windows, предназначенные для работы с каналами MailSlot, которые сведем в таблицу 1.

Таблица 1

Функции для работы с каналами MailSlot

|  |  |  |
| --- | --- | --- |
| № п/п | Функция | Назначение |
|  | CreateMailslot | Создание канала MailSlot |
|  | CloseHandle | Закрытие хэндла канала |
|  | GetMailslot | Определение текущего состояния канала Mailslot. |
|  | SetMailslot | Изменение времени ожидания для канала Mailslot после его создания |

#### Создание канала Mailslot

Канал Mailslot создается серверным процессом с помощью специально предназначенной для этого функции CreateMailslot().

После создания серверный процесс получает хэндл канала Mailslot. Пользуясь этим хэндло, сервер может читать сообщения, посылаемые в канал клиентскими процессами.

Однако сервер не может выполнять над каналом Mailslot операцию записи, так как этот канал предназначен только для односторонней передачи данных - от клиента к серверу.

Синтаксис функции следующий

HANDLE CreateMailslot (LPCTSTR lpName, DWORD nMaxMsgSize, DWORD lReadTimeout, LPSECURITY\_ATTRIBUTES lpSecurityAttributes)

Параметры:

lpName – указатель на строку имени канала Mailslot.

nMaxMsgSize – максимальный размер сообщения.

lReadTimeout – время ожидания для чтения.

lpSecurityAttributes – указатель на атрибуты защиты

Возвращаемое значение. При успешном выполнении – хэндл созданного канала Mailslot, при ошибке – INVALID\_HANDLE\_VALUEКод ошибки уточняется при вызове функции GetLastError()

Через первый параметр lpName передаются функции CreateMailslot()указатель на строку символов с именем канала Mailslot. Эта строка имеет следующий вид: \\.\mailslot\[Путь]ИмяКанала

В этом имени путь является необязательной компонентой. Тем не менее, вы можете указать его аналогично тому, как это делается для файлов. Что же касается имени канала Mailslot, то оно задается аналогично имени канала Pipes.

Второй параметр nMaxMsgSize определяет максимальный размер сообщений, передаваемых через создаваемый канал Mailslot.

Здесь можно указать нулевое значение, при этом размер сообщений не будет ограничен. Есть, однако, одно исключение - размер широковещательных сообщений, передаваемых всем рабочим станциям и серверам домена не должен превышать 400 байт.

С помощью третьего параметра lReadTimeout серверное приложение может задать время ожидания для операции чтения в миллисекундах, по истечении которого функция чтения вернет код ошибки.

Если указать в этом параметре значение MAILSLOT\_WAIT\_FOREVER, ожидание будет бесконечным.

Четвертый параметр lpSecurityAttributes задает адрес структуры защиты, который мы в наших приложениях будем указывать как NULL.

Пример 1

Представим использование функции CreateMailslot()в серверном приложении:

LPSTR lpszMailslotName = "\\\\.\\mailslot\\$MailslotName$";

hMailslot = CreateMailslot(lpszMailslotName, 0, MAILSLOT\_WAIT\_FOREVER, NULL);

В этом примере задан максимальный размер сообщения, поэтому на эту величину нет ограничений (кроме ограничения в 400 байт для сообщений, передаваемых всем компьютерам домена в широковещательном режиме).

Время ожидания указано как MAILSLOT\_WAIT\_FOREVER, поэтому функции, работающие с данным каналом Mailslot будут работать в блокирующем режиме.

#### 

#### Открытие канала Mailslot

Прежде чем приступить к работе с каналом Mailslot, клиентский процесс должен его открыть.

Для выполнения этой операции следует использовать функцию CreateFile(), например, так:

LPSTR lpszMailslotName = "\\\\.\\mailslot\\$MailslotName$";

hMailslot = CreateFile (lpszMailslotName, GENERIC\_WRITE, FILE\_SHARE\_READ, NULL, OPEN\_EXISTING, 0, NULL);

Здесь в качестве первого параметра функции CreateFile() передается имя канала Mailslot

Заметим, что возможно открыть канал Mailslot, созданный на другой рабочей станции в сети. Для этого строка имени канала, передаваемая функции CreateFile(), должна иметь следующий вид: \\ИмяРабочейСтанции\mailslot\[Путь]ИмяКанала

Можно открыть канал для передачи сообщений всем рабочим станциям заданного домена. Для этого необходимо задать имя по следующему образцу: \\ИмяДомена\mailslot\[Путь]ИмяКанала

Для передачи сообщений одновременно всем рабочим станциям сети первичного домена имя задается следующим образом: ь\\\*\mailslot\[Путь]ИмяКанала

В качестве второго параметра функции CreateFile() передается константа GENERIC\_WRITE

Эта константа определяет, что над открываемым каналом будет выполняться операция записи.

Напомним, что клиентский процесс может только посылать сообщения в канал Mailslotно не читать их оттуда. Чтение сообщений из канала Mailslot - задача для серверного процесса.

Третий параметр указан как FILE\_SHARE\_READ, и это тоже необходимо, так как сервер может читать сообщения, посылаемые одновременно несколькими клиентскими процессами.

Обратите также внимание на константу OPEN\_EXISTING. Она используется потому, что функция CreateFile()открывает существующий канал, а не создает новый.

#### Запись сообщений в канал Mailslot

Запись сообщений в канал Mailslot выполняет клиентский процесс, вызывая для этого функцию WriteFile. С этой функцией мы уже имели дело. Например,

HANDLE hMailslot;

char szBuf[512];

DWORD cbWritten;

WriteFile (hMailslotszBuf, strlen(szBuf) + 1, &cbWritten, NULL);

В качестве первого параметра этой функции необходимо передать хэндл канала Mailslotполученный от функции CreateFile().

Второй параметр определяет адрес буфера с сообщением, третий - размер сообщения. В нашем случае сообщения передаются в виде текстовой строки, закрытой двоичным нулем, поэтому для определения длины сообщения была использована функция strlen.

#### 

#### Чтение сообщений из канала Mailslot

Серверный процесс может читать сообщения из созданного им канала Mailslot при помощи функции ReadFile(), как это показано ниже:

HANDLE hMailslot char szBuf[512]; DWORD cbRead;

ReadFile (hMailslotszBuf, 512, &cbRead, NULL);

Через первый параметр функции ReadFile передается хэндл созданного ранее канала Mailslot, полученный от функции CreateMailslot(). Второй и третий параметры задают, соответственно, адрес буфера для сообщения и его размер.

Заметим, что перед выполнением операции чтения следует проверить состояние канала Mailslot.

Если в нем нет сообщений, то функцию ReadFile()вызывать не следует. Для проверки состояния канала вы должны воспользоваться функцией GetMailslotInfo().

#### Определение состояния канала Mailslot

Серверный процесс может по его хэндлу с помощью функции GetMailslotInfo(), синтаксис которой следующий

BOOL GetMailslotInfo (HANDLE hMailslot, LPDWORD lpMaxMessageSize, LPDWORD lpNextSize, LPDWORD lpMessageCount, LPDWORD lpReadTimeout)

Параметры:

hMailslot –хэндл канала Mailslot.

lpMaxMessageSize – адрес максимального размера сообщения.

lpNextSize – адрес размера следующего сообщения.

lpMessageCount – адрес количества сообщений.

lpReadTimeout – адрес времени ожидания.

Возвращаемое значение. При успешном выполнении – TRUE, при ошибке - FALSE.

Через параметр hMailslot функции передается идентификатор канала Mailslot, состояние которого необходимо определить.

Остальные параметры задаются как указатели на переменные типа DWORD, в которые будут записаны параметры состояния канала Mailslot.

В переменную, адрес которой передается через параметр lpMaxMessageSize, после возвращения из функции GetMailslotInfo() будет записан максимальный размер сообщения.

Можно использовать это значение для динамического получения буфера памяти, в который это сообщение будет прочитано функцией ReadFile()

В переменную, адрес которой указан через параметр lpNextSize, записывается размер следующего сообщения, если оно есть в канале. Если же в канале больше нет сообщений, в эту переменную будет записана константа MAILSLOT\_NO\_MESSAGE

С помощью параметра lpMessageCount можно определить количество сообщений, записанных в канал клиентскими процессами.

Если это количество равно нулю, то не следует вызывать функцию ReadFile()для чтения несуществующего сообщения.

И, наконец, в переменную, адрес которой задается в параметре lpReadTimeout, записывается текущее время ожидания, установленное для канала (в миллисекундах).

Если не нужна вся информация, которую можно получить с помощью функции GetMailslotInfo(), некоторые из ее параметров (кроме, разумеется, первого) можно указать как NULL.

Пример 2. Демонстрация использования функции GetMailslotInfo():

BOOL fReturnCode; DWORD cbMessages;DWORD cbMsgNumber;

fReturnCode = GetMailslotInfo(hMailslot, NULL, &cbMessages, &cbMsgNumber, NULL);

#### Изменение состояния канала Mailslot

С помощью функции SetMailslotInfo() серверный процесс может изменить время ожидания для канала Mailslot уже после его создания.

Синтаксис функции SetMailslotInfo() следующий:

BOOL SetMailslotInfo (HANDLE hMailslot, DWORD lpReadTimeout)

Параметры:

hMailslot –хэндл канала Mailslot.

lpReadTimeout – адрес времени ожидания.

Возвращаемое значение. При успешном выполнении – TRUE, при ошибке - FALSE.

Через параметр hMailslot функции SetMailslot Info() передается идентификатор канала Mailslot, для которого нужно изменить время ожидания.

Новое значение времени ожидания в миллисекундах задается через параметр dwReadTimeout. Также можно указать здесь константы 0 или MAILSLOT\_WAIT\_FOREVER.

В первом случае функции, работающие с каналом, вернут управление немедленно, во втором - будут находиться в состоянии ожидания до тех пор, пока не завершится выполняемая операция.