// **-- создать почтовый ящик**

// **Назначение:** функция предназначена для создания почтового

// ящика

**HANDLE CreateMailslot**

**(**

**LPCTSTR pname,**  // [in] символическое имя ящика

**DWORD maxms,** // [in] максимальная длина сообщения

**DWORD timeo,** //[in] интервал ожидания

**LPSECURITY\_ATTRIBUTES sattr** // [in] атрибуты безопасности

**);**

// **Код возврата:** в случае успешного завершения функция

// возвращает дескриптор почтового ящика, иначе

// значение **INVALID\_HANDLE\_VALUE**

// **Примечание: pname** - указывает на строку именем канала в

// **локальном формате**;

// **timeo** – параметр устанавливает время ожидания

// cобщения функцией **ReadFile;** для задания бесконечного

// ожидания, следует установить значение

// **MAILSLOT\_WAIT\_FOREVER;**

// **sattr –** для установки атрибутов безопасности

// по умолчанию следует установить значение **NULL**

В этой главе рассматривается еще один IPC – механизм, поддерживаемый операционной системой Windows и имеющий название ***Mailslots*** ***(почтовый ящик)***. Также как и Named Pipe механизм Mailslots может быть использован для обмена данными между распределенными в локальной сети процессами.

**5.2. Назначение и состав интерфейса Mailslot**

Почтовым ящиком (Mailslot) называется объект ядра операционной системы, который обеспечивает передачу данных от процессов-клиентов к процессам-серверам, выполняющимся на компьютерах в одной локальной сети. Процесс, создающий почтовый ящик называется ***сервером почтового ящика***. Процессы, которые связываются с почтовым ящиком, называются ***клиентами почтового*** ящика.

**\\.\mailslot\xxxxx**

где: **точка (.) -** обозначает локальный компьютер;

**mailslot -** фиксированное слово;

**xxxxx -** имя почтового ящика

,...

// **-- 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**

// **-- открыть почтовый ящик**

// **Назначение:** функция предназначена для подключения клиента

// к почтовому ящику

**HANDLE CreateFile**

**(**

**LPCTSTR mname,** // [in] символическое имя почтового ящика

**DWORD accss,** // [in] чтение или запись

**DWORD share,** // [in] режим совместного использования

**LPSECURITY\_ATTRIBUTES sattr** // [in] атрибуты безопасности

**DWORD oflag,** // [in] флаг открытия почтового ящика

**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 **mname** указывается в локальном,

// сетевом или доменном формате: в зависимости от

// способа применения;

// - параметр **accss** должен принимать значение

// **GENERIC\_WRITE**

// - параметр **share** можетпринимать 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** всегдаустанавливаетсяв

// **OPEN\_EXISTING**(открытие существующего ящика);

// - значение параметра **aflag** можно установить в **NULL**,

// что определяет значения флагов и атрибутов по

// умолчанию или установить **FILE\_ATTRIBUTE\_NORMAL**;

// - значение параметра **exten** следует становить в **NULL**

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

**HANDLE hM;** // дескриптор почтового ящика

**DWORD wb;** // длина записанного сообщения

**char wbuf[] = "Hello Mailslot”;** //буфер вывода

**try**

**{**

**if ((hM = CreateFile("\\\\isit301\\mailslot\\myslot",//** символическое имя почтового ящика

**GENERIC\_WRITE,** // будем писать в ящик

**FILE\_SHARE\_READ,** // разрешаем одновременно читать

**NULL, //атрибут безопасности**

**OPEN\_EXISTING,** // открытие сущест.ящикь

**NULL, //атр безопасности**

**NULL)) //атр безопасности**

**== INVALID\_HANDLE\_VALUE)**

**throw "CreateFileError";**

**if(!WriteFile(hM,**

**wbuf,** // буфер

**sizeof(wbuf),** // размер буфера

**&wb,** // записано

**NULL))**

**throw "ReadFileError";**

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

**}**

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

// **-- 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**

// **-- 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**

|  |  |
| --- | --- |
| **DisconnectNamedPipe** | Закончить обмен данными |

Closehandle—zakryvaet diskriptor