СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

ЛЕКЦИЯ № 9

ПРЕПОДАВАТЕЛЬ: ХУСТОЧКА А.В.

ПОЧТОВЫЕ ЯЩИКИ

- Почтовым ящиком называется объект ядра операционной системы, который обеспечивает передачу сообщений от процессов-клиентов к процессам серверам, выполняющимся на компьютерах в пределах локальной сети.
- Процесс, который создает почтовый ящик, называется сервером почтового ящика.
- Процессы, которые связываются с именованным почтовым ящиком, называются клиентами почтового ящика.
- Характеристики почтовых ящиков:
  - имеют имя, которое используется клиентами для связи с почтовыми ящиками;
  - направление передачи данных - от клиента к серверу;
  - передача данных осуществляется сообщениями;
  - обмен данными может быть как синхронным, так и асинхронным.

ПОЧТОВЫЕ ЯЩИКИ

- Передача данных осуществляется только от клиента к серверу.
- Несколько серверов могут создать почтовые ящики с одинаковыми именами, и тогда все сообщения, которые посылает клиент в такой почтовый ящик, будут получать все серверы этого почтового ящика.
- Однако в этом случае длина сообщения должна быть меньше 425 байт, так как сообщения передаются дейтаграммами. Таким образом, можно сказать, что почтовые ящики обеспечивают однонаправленную связь типа "многие-ко-многим". При этом доставка сообщения от клиента к серверам почтового ящика не подтверждается системой. Заметим также, что операционные системы семейства Windows NT не поддерживают передачу сообщений длиной 425 и 426 байт.

ПОЧТОВЫЕ ЯЩИКИ

- Если длина сообщения меньше, чем 425 байт, то такое сообщение передается как дейтаграмма. Дейтаграмма представляет собой небольшой пакет с передаваемым по сети сообщением, который содержит также информацию об отправителе и получателе сообщения. Дейтаграмма рассылается всем серверам данного почтового ящика. Так как размер пакета небольшой, то дейтаграммы рассылаются быстро, но нет гарантии доставки сообщения, так как в дейтаграмме не хранится информация, поддерживающая контроль доставки.
- Если же длина сообщения больше 426 байт, то такие сообщения могут передаваться только от одного клиента к одному серверу, используя при этом SMB (Server Message Block) - протокол передачи данных по сети. При этом отметим, что длина сообщения, передаваемого в почтовый ящик, не может превышать 64 Кбайт.
- Почтовые ящики можно рассматривать как псевдофайлы, расположенные в оперативной памяти компьютера. Поэтому для доступа к почтовым ящикам используются те же функции, что и для доступа к обычным файлам.

ПОРЯДОК РАБОТЫ С ПОЧТОВЫМИ ЯЩИКАМ

Порядок работы с почтовыми ящиками, который и будет использоваться в дальнейшем:

- Создание почтового ящика сервером.
- Соединение клиента с почтовым ящиком.
- Обмен данными через почтовый ящик.
- Закрытие почтового ящика клиентом и сервером.

СОЗДАНИЕ ПОЧТОВОГО ЯЩИКА

- Создаются почтовые ящики процессом-сервером при помощи функции CreateMailslot, которая имеет следующий прототип:

```
HANDLE
WINAPI
CreateMailslotW(
_In_     LPCWSTR lpName,         // имя почтового ящика
_In_     DWORD nMaxMessageSize, // максимальная длина сообщения
_In_     DWORD lReadTimeout,    // интервал ожидания
_In_opt_ LPSECURITY_ATTRIBUTES lpSecurityAttributes // атрибуты безопасности
);
```

- В случае успешного завершения эта функция вернет дескриптор почтового ящика, а в случае неудачи — значение INVALID_HANDLE_VALUE.

СОЗДАНИЕ ПОЧТОВОГО ЯЩИКА

- **Параметр lpName** указывает на строку, которая должна иметь вид: \\.\mailslot\mailslot_name
- Здесь символ "." обозначает локальную машину, так как новый почтовый ящик всегда создается на локальной машине, слово mailslot — фиксировано, а mailslot_name обозначает имя почтового ящика, которое задается пользователем и нечувствительно к верхнему и нижнему регистрам.
- **Параметр dwMaxMessageSize** задает максимальную длину сообщения в байтах, которое может быть записано в почтовый ящик.
- **Параметр dwReadTimeout** задает в миллисекундах временной интервал, в течение которого функция ReadFile ждет поступления сообщения в почтовый ящик. Если в этом параметре установлено значение 0, то в случае отсутствия в почтовом ящике сообщения функция немедленно возвращает управление. Для задания бесконечного времени ожидания в этом параметре нужно установить значение MAILSLOT_WAIT_FOREVER.
- Несколько процессов могут создать почтовые ящики с одним и тем же именем. В этом случае сообщение, посланное клиентом, доставляется не только в почтовый ящик одного процесса, а также в почтовые ящики всех таких процессов при условии, что они работают на компьютерах внутри одного домена. Режим доставки сообщений зависит от режима открытия почтового ящика клиентом.

СОЕДИНЕНИЕ КЛИЕНТА С ПОЧТОВЫМ ЯЩИКОМ

- Для установления связи с почтовым ящиком клиент использует функцию CreateFile, которая имеет следующий прототип:

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

- В случае успешного завершения эта функция возвращает дескриптор почтового ящика, а в случае неудачи — значение INVALID_HANDLE_VALUE.

СОЕДИНЕНИЕ КЛИЕНТА С ПОЧТОВЫМ ЯЩИКОМ

- **Параметр lpFileName** должен указывать на имя почтового ящика, которое может быть задано в одном из следующих форматов:
  - почтовый ящик на данном локальном компьютере: \\.\mailslot\имя_почтового_ящика
  - почтовый ящик на компьютере с указанным именем: \\имя_компьтера\mailslot\имя_почтового_ящика
  - почтовый ящик в домене с указанным именем: \\имя_домена\mailslot\имя_почтового_ящика
  - почтовый ящик в первичном (т.е. данном) домене системы: *\mailslot\имя_почтового_ящика
- В первом случае сообщения будут доставляться только в почтовые ящики с заданным именем, которые расположены на локальной машине.
- Во втором случае сообщения будут доставляться в почтовые ящики с заданным именем, расположенные на компьютере с указанным именем.
- В третьем случае сообщения будут доставляться в почтовые ящики с заданным именем, которые созданы внутри домена с указанным именем.
- В четвертом случае сообщения будут доставляться в почтовые ящики с заданным именем, которые созданы внутри первичного домена системы.

СОЕДИНЕНИЕ КЛИЕНТА С ПОЧТОВЫМ ЯЩИКОМ

- **Параметр dwDesiredAccess** должен иметь значение GENERIC_WRITE, которое разрешает запись в почтовый ящик.
- **Параметр dwShareMode** определяет режим совместного использования почтового ящика и может принимать любую комбинацию из следующих значений:
  - FILE_SHARE_READ — разрешает совместное чтение из почтового ящика;
  - FILE_SHARE_WRITE — разрешает совместную запись в почтовый ящик.
- **Параметр lpSecurityAttributes** задает атрибуты безопасности почтового ящика. Пока этот параметр будем устанавливать в NULL.
- Для почтового ящика **параметр dwCreationDisposition** должен быть равен значению OPEN_EXISTING, т.к. клиент всегда открывает существующий почтовый ящик.
- Для почтового ящика **параметр dwFlagsAndAttributes** можно задать равным 0, что определяет флаги и атрибуты по умолчанию, или установить в этом параметре значение FILE_ATTRIBUTE_NORMAL.
- **Параметр hTemplateFile** при работе с почтовыми ящиками не используется, поэтому в нем устанавливается значение NULL.

ОБМЕН ДАННЫМИ ЧЕРЕЗ ПОЧТОВЫЙ ЯЩИК

- Для обмена данными через почтовый ящик используются обычные функции доступа к файлу WriteFile и ReadFile.
- Процесс-клиент записывает данные в почтовый ящик при помощи функций writeFile, а процесс-сервер читает данные из почтового ящика, используя функции ReadFile.
- Конечно, процесс-сервер также может записывать сообщения в почтовый ящик, но это имеет смысл лишь в том случае, если в домене расположено несколько почтовых ящиков с одинаковыми именами, которые были созданы разными процессами. Тогда одинаковые сообщения получают все почтовые ящики с одинаковыми именами.

ПОЛУЧЕНИЕ ИНФОРМАЦИИ О ПОЧТОВОМ ЯЩИКЕ

- Для получения информации о характеристиках почтового ящика используется функция GetMailslotInfo, которая имеет следующий прототип:

```
BOOL
WINAPI
GetMailslotInfo(
_In_      HANDLE hMailslot,        // дескриптор почтового ящика
_Out_opt_ LPDWORD lpMaxMessageSize, // максимальная длина сообщения
_Out_opt_ LPDWORD lpNextSize,      // длина следующего сообщения
_Out_opt_ LPDWORD lpMessageCount, // количество сообщений
_Out_opt_ LPDWORD lpReadTimeout   // интервал ожидания сообщения
);
```

- В случае успешного завершения эта функция возвращает ненулевое значение, а в случае неудачи — NULL.

ПОЛУЧЕНИЕ ИНФОРМАЦИИ О ПОЧТОВОМ ЯЩИКЕ

- В **параметре hMailslot** должен быть установлен дескриптор почтового ящика, который был возвращен функцией CreateMailslot.
- **Параметр lpMaxMessageSize** должен указывать на переменную типа DWORD, в которую функция GetMailslotInfo поместит максимальную длину сообщения, которое может быть записано в почтовый ящик. Если это значение не нужно, то этот параметр может быть установлен в NULL.
- **Параметр lpNextSize** должен указывать на переменную типа DWORD, в которую функция GetMailslotInfo поместит длину следующего сообщения в почтовом ящике. Если в почтовом ящике нет сообщений, то в этот параметр функция запишет значение MAILSLOT_NO_MESSAGE. Если значение длины последнего сообщения не нужно, то параметр lpNextSize может быть установлен в NULL.

ПОЛУЧЕНИЕ ИНФОРМАЦИИ О ПОЧТОВОМ ЯЩИКЕ

- **Параметр lpMessageCount** должен указывать на переменную типа DWORD, в которую функция GetMailslotInfo поместит количество сообщений, находящихся в почтовом ящике. Если это значение не нужно, то этот параметр может быть установлен в NULL.
- **Параметр lpReadTimeout** должен указывать на переменную типа DWORD, в которую функция GetMailslotInfo поместит целое число без знака, обозначающее временной интервал в миллисекундах. Если почтовый ящик пуст, то в течение этого интервала функция ReadFile будет ждать, пока процесс-клиент не запишет сообщение в почтовый ящик. Если это значение не нужно, то в этом параметре может быть установлено значение NULL.

ИЗМЕНЕНИЕ ВРЕМЕНИ ОЖИДАНИЯ СОБЫТИЯ

- Для изменения времени ожидания сервером сообщения от клиента используется функция SetMailslotInfo, которая имеет следующий прототип:

```
BOOL
WINAPI
SetMailslotInfo(
_In_ HANDLE hMailslot,  // дескриптор почтового ящика
_In_ DWORD lReadTimeout // интервал ожидания сообщения
);
```

- В случае успешного завершения эта функция возвратит ненулевое значение, а в случае неудачи — NULL.
- **Параметр dwReadTimeout** задает в миллисекундах новый временной интервал, в течение которого функция ReadFile ждет поступления сообщения в почтовый ящик. Если в этом параметре устанавливается значение 0, то в случае отсутствия в почтовом ящике сообщения функция немедленно возвращает управление. Для задания бесконечного времени ожидания в этом параметре нужно установить значение MAILSLOT_WAIT_FOREVER.