

Использование файлов, отображаемых в память

Отображение файлов в память позволяет процессу интерпретировать содержимое файла так, как будто оно является блоком памяти в его адресном пространстве. Процесс может использовать простые операции для чтения и модификации содержимого файла. Когда два или более процесса получают доступ к одному и тому же отображенному в память файлу (данным), каждый процесс получает указатель на память в своем собственном адресном пространстве и может читать или модифицировать содержимое файла. Процессы должны использовать синхронизирующие объекты, такие как семафоры, для того чтобы предотвратить разрушение данных в многозадачной среде.

Можно также использовать специальный вид отображения файлов, называемый именованной **разделяемой памятью** (*named shared memory*). Если создаваемый объект отображения файла в память во время его создания определить как системный файл подкачки, то он будет интерпретироваться как блок разделяемой памяти. Разные процессы могут получить доступ к одному и тому же блоку памяти путем открытия одного и того же объекта.

Отображение файла в память также обеспечивает поддержку атрибутов безопасности ОС, которая может помочь предохранить данные от несанкционированного доступа. Отображение файла в память может быть использовано только для взаимодействия между процессами на одном компьютере, но не в сети.

Таким образом, отображаемые в память файлы предоставляют возможность нескольким процессам получать совместный доступ к данным, однако процессы должны обеспечивать синхронизацию доступа к данным.

Имеется два типа каналов (pipes) для двустороннего межпроцессного взаимодействия: анонимные (anonymous) и именованные (named).

Неименованные каналы (Anonymous Pipe)

позволяют родственным (related) процессам передавать информацию друг другу. Обычно такой канал используется для перенаправления стандартного ввода или вывода дочернего процесса так, что он может обмениваться данными со своим родительским процессом. Для обмена данными в двух направлениях (дуплексные операции) необходимо создавать два анонимных канала. Родительский процесс записывает данные в один канал с использованием своего дескриптора записи (write handle), в то время как дочерний процесс читает данные из этого канала с использованием своего дескриптора чтения (read handle). Аналогичным образом, дочерний процесс записывает данные в другой канал, а родительский читает эти данные. Анонимные каналы не могут быть использованы в сети или неродственными процессами.

Именованные каналы (Named Pipes)

используются для передачи данных между процессами, которые не являются родственными, и между процессами на разных компьютерах. Обычно, процесс-сервер создает именованный канал с широко известным именем или таким именем, которое предназначено для связи с клиентами. Процесс-клиент, который знает это имя, может открыть именованный канал в соответствии с ограничениями, наложенными сервером. После того как оба процесса присоединились к каналу, они могут обмениваться данными.

Резюме. Неименованные каналы предоставляют эффективный способ для перенаправления стандартного ввода или вывода дочернего процесса на том же компьютере. Именованные каналы предоставляют простой программный интерфейс для передачи данных между процессами, которые находятся на одном компьютере или на разных.

Почтовые ящики (MailSlots)

Подобны именованным каналы, но предоставляют более простой однонаправленный интерфейс. Процесс-сервер может создать канал 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 читает переданные ему данные. Возможна также широковещательная передача информации клиентом всем серверам домена. Почтовый ящик создается функцией CreateMailslot, причем используется имя вида `\\.\mailslot\[<path>]\<slotname>`, где собственно имя ящика может включать символы '\', не имея соответствия в реальной структуре каталогов (т.н. *псевдодиректории*). Для доступа к ящику файловыми функциями его надо открыть функцией CreateFile с использованием следующих вариантов имени файла:

`\\.\mailslot\<slot_name>` – локальный ящик;

`\\<computer_name>\mailslot\<slot_name>` – удаленный ящик;

`\\<domain_name>\mailslot\<slot_name>` – используя доменное имя;

`*\mailslot\<slot_name>` – используя первичный (primary) домен системы.

В ящик, открытый через доменное имя, нельзя записывать более 400 байт в один прием.

Если ящик будет использоваться более чем одним процессом (владельцем), то CreateFile должен задавать флаг доступа FILE_SHARE_READ.

Созданный ящик существует до тех пор, пока существует процесс-владелец, либо пока владелец или его потомок не удалят его оператором вызова CLOSE_HANDLE.

Резюме. Почтовые ящики предоставляют простой способ для приложений посылать и получать сравнительно короткие сообщения. Они также предоставляют простую возможность рассылать сообщения всем компьютерам домена.