

Драйвер режима ядра для поддержки межпроцессного обмена

Система межпроцессного обмена (далее IPC) представлена библиотекой режимой пользователя и драйвером режима ядра. Драйвер представляет собой драйвер «в стиле NT». Он не является драйвером устройств. Драйвер использует общедоступные функции ядра, поэтому работает на всей линейке операционных систем Windows NT, начиная с Windows 2000.

Основное назначение – это образовывать некий «мост» между различными процессами. Основное преимущество данного драйвера в составе IPC в том, что он позволяет обмениваться сообщениями незаметно для операционной системы. Как известно, основой для межпроцессного обмена в различных широко известных библиотеках являются либо сообщения окон операционной системы, либо использование сетевых сокетов, либо разделяемой памяти (проблематично в Windows Vista). Все эти способы могут контролироваться операционной системой и блокироваться на уровне пользователя. Например, при попытке использования IPC на основе сокетов, возможно возникновение диалога о разрешении использования сети приложением, использующий сокет. Сообщения окон, начиная с Windows Vista, вообще могут не доходить до адресата, хотя результат выполнения будет возвращаться как «все отлично». Это происходит из-за различия «уровней» привилегий даже для приложений, работающих под одной и той же учетной записью. Использование разделяемой памяти становится проблемой при попытке обмена между приложениями, работающими в различных сеансах. Основываясь на выше описанных недостатках стандартных средств операционной системы, было решено разработать совершенно независимую систему. Код, выполняющий высокоскоростные, низкоуровневые операции, было предложено вынести в код драйвера, а высокоуровневую логику решено было поместить в библиотеку режима пользователя.

Логически, код драйвера можно разделить на несколько логических частей, каждая из которых работает с различными «псевдо-объектами». Основными объектами для манипуляции стали:

- Объект сообщения
- Объект процесса
- Объект глобального адреса

Для манипуляции объектами была создана среда (набор API) подобно runtime C++. Для ускорения работы кода память, используемая для объектов, выделяется при инициализации системы. Недостаток этого метода в ограниченном количестве одновременно существующих объектов, но этот недостаток легко перекрывается скоростью работы. Выделение памяти единовременно также решает извечную проблему «утечек».

Память, выделяемая при старте, также делится на условные части:

- Собственно память для объектов
- Память для канального обмена
- Память для временного хранения данных (набор слотов)

Таким образом, условную схему драйвера IPC можно изобразить так:

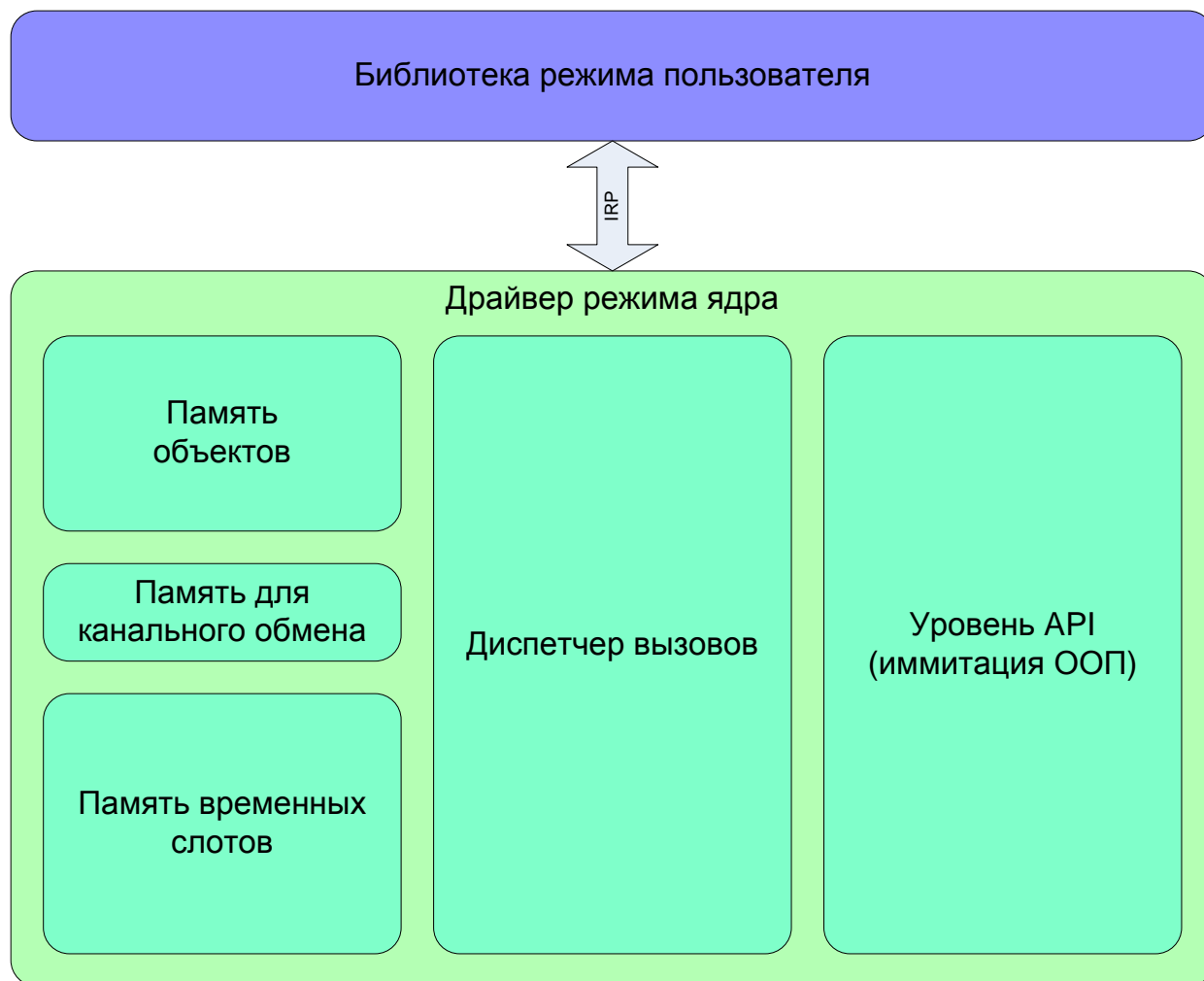


Рис. Общая схема IPC

Объекты

Объект сообщения – представляет собой базовый набор данных: код сообщения, первый параметр, второй параметр (подобно оконному сообщению), адрес назначения, адрес отправителя (возможно фиктивный), тип сообщения (синхронное, асинхронное) и т.д.

Объект процесса – это набор данных описывающий текущий процесс операционной системы в системе IPC: это и уникальный идентификатор IPC адреса процесса и синхронизирующие объекты, и очередь сообщений и т.д.

Объект глобального адреса – это специальная структура, хранящая описание адреса IPC уникального в пределах данной операционной системы. Следует сказать, что раз существует глобальный адрес, значит существует и локальный адрес в пределах данного процесса. Таким образом, суммируя, можно сказать о наличии двух адресаций: вертикальной – в пределах процесса, и горизонтальной – в пределах системы:

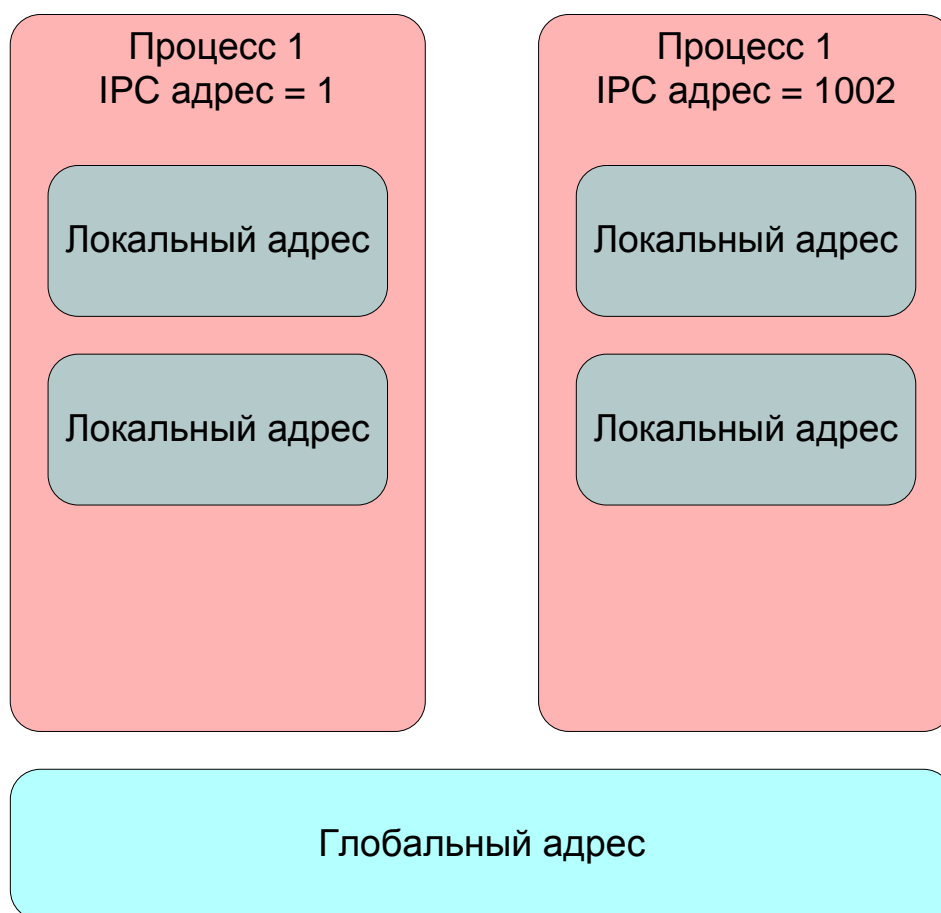


Рис. Адресация IPC

Таким образом, для формирования адреса пункта назначения необходимо указать тип адресации (глобальная или нет), адрес процесса и адрес внутри процесса.

Базовые сообщения не позволяют передавать большие объемы данных (всего лишь два параметра). Для этого было решено реализовать обмен данными на базе сообщений. Обмен происходит через разделяемую память, расположенную в драйвере. Сами данные накапливаются в режиме пользователя в так называемых канальных контейнерах. То есть, для передачи блока данных вы резервируете канал, передаете данные в него, посылаете сообщение обработчику, потом освобождаете канал и так далее. Но этот способ чреват взаимоблокировками, поэтому его необходимо использовать с осторожностью.

Есть более свободный способ передачи данных, но есть ограничение на их объем (порядка 500 кБайт за одну транзакцию). Этот способ использует память слотов. Этот механизм представляет собой некое подобие почтовой системы у людей. Отправитель данных кладет их в свободный слот и указывает интервал, в течение которого эти данные должны находиться в слоте. Как только интервал истечет, данные будут удалены потоком очистки ресурсов. Такая организация позволяет сохранять работоспособность системы в случае критического завершения работы процесса получателя либо отправителя: представьте, вы положили в слот данные и указали интервал в 5 секунд, но вдруг, ваш процесс завершается и вы не успеваете изъять данные из слота – ничего страшного, система чистки сделает это за вас через 5 секунд.

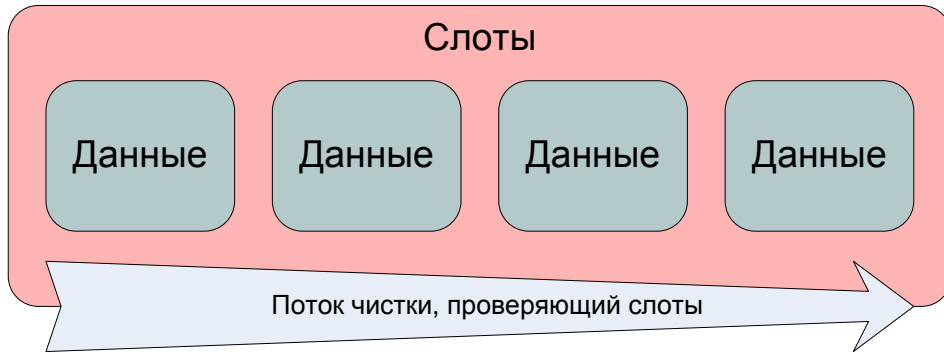


Рис. Система слотов временного хранения

Иногда возникает необходимость посылки сообщения одновременно нескольким адресатам. Для этого в системе IPC предусмотрена широковещательная рассылка. Для этого необходимо указать лишь локальный адрес назначения, сообщение будет доставлено во все процессы, зарегистрированные в системе IPC.

Организация работы драйвера основывается на очередях сообщений. Если есть сообщение для обработки, то оно ассоциируется с объектом процесса. В свою очередь объект процесса используется обрабатывающим потоком режима пользователя, он сканирует очередь и ищет новые сообщения. Уже обработанные сообщения либо удаляются сразу из очереди (если асинхронные), либо помечаются как готовые либо недошедшие до адресата (если процесс назначения уже не существует).

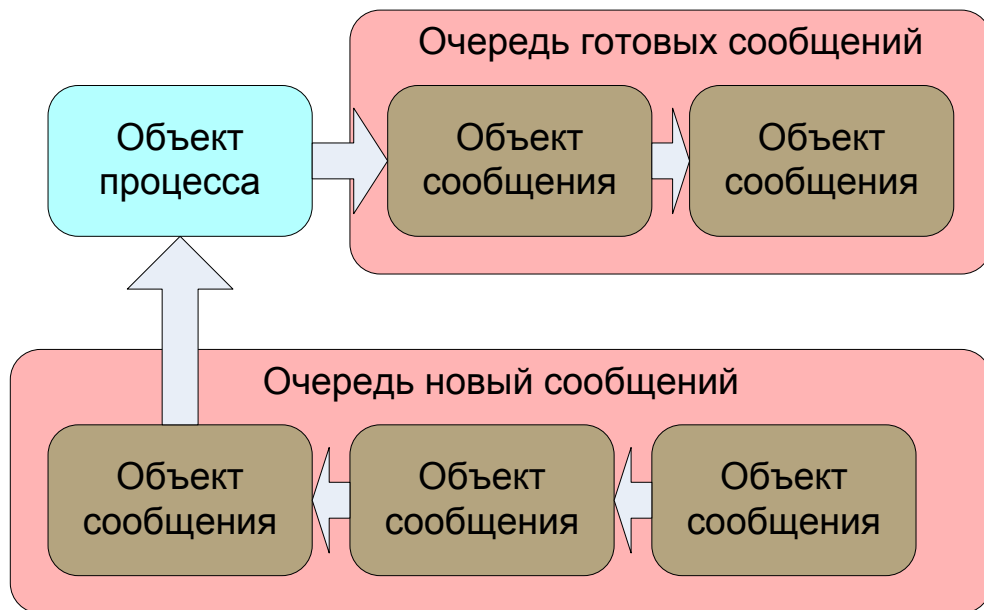


Рис. Обработка сообщений

Драйвер системы хранения настроек

Система хранения настроек работает подобно реестру операционной системы. Она предназначена для хранения элементарных типов данных:

- 1,2,4,8 байтовых чисел
- Строковых и мультистроковых значений
- Бинарных данных

Драйвер реализует хранение древовидного списка имен и путей и одновременный сброс данных на жесткий диск. Подобная схема используется и в операционной системе. Различие состоит в том, что наша реализация не использует систему описателей. Для работы со значением вам необходимо указать лишь путь к значению и его тип. Оптимизация работы драйвера достигается использованием уже имеющейся памяти для хранения нового значения если его размер не превышает размер уже выделенного блока. При достижении предела памяти происходит дефрагментация всего реестра с выделением блока памяти большим на определенное значение.

Скорость поиска нужного значения достигается использованием алгоритма двоичного поиска.

Для поддержки шифрование имеется возможность встраивания упаковки «налету». Сам ключ может передаваться как составляющая полного имени элемента.

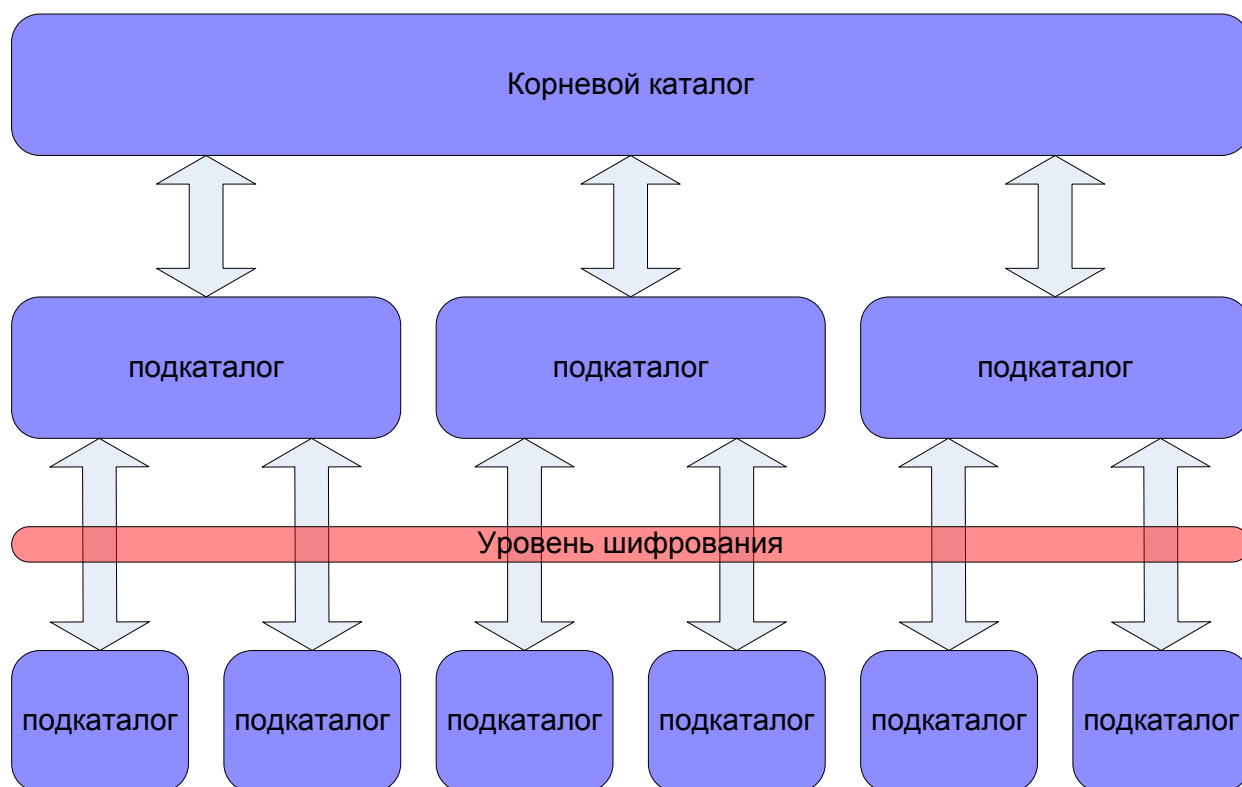


Рис. Общая схема драйвера системы хранения настроек

Драйвер Network Redirector

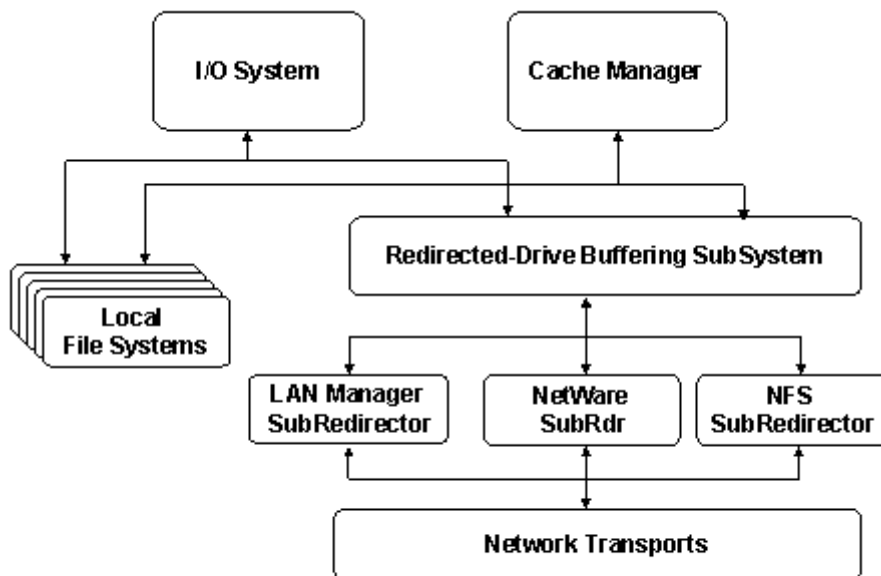


Рис. Архитектура мини-редиректора в Windows 2000.

Network Redirector – это программные компоненты, установленные на компьютере клиента, которые используются для того, чтобы предоставить пользователю доступ к файлам и другим ресурсам (принтеры, например) на удаленной системе. Network Redirector отправляет (или переадресовывает) запросы от локальных приложений на операции над файлами на удаленный сервер, где эти запросы обрабатываются. Network Redirector получает ответы от удаленного сервера, которые затем возвращаются локальному приложению. Network Redirector представляет на системе клиента удаленные файлы и ресурсы как локальные ресурсы и позволяет их использовать теми же самыми способами.

Network Redirector пытается сделать доступ к удаленным ресурсам настолько прозрачным, насколько возможно для локального приложения-клиента.

В Windows 2000 была введена новая модель драйвера (часто называемая архитектурой мини-редиректора, или rdr2) для систем Network Redirector. Вместо того, чтобы в каждом драйвере повторно не реализовывать сложный код для буферизации и взаимодействия с менеджером ввода/вывода, менеджера кэша, этот большой блок кода был вынесен и сделан общедоступным для всех драйверов Network Redirector.

Этот общедоступный код называют Redirected Drive Buffering SubSystem (RDBSS).

Нами был разработан драйвер мини-редиректора, для предоставления пользователю доступа к FTP и WebDav ресурсам. Драйвер разрабатывался согласно всех требований и рекомендаций архитектуры мини-редиректора, поэтому является совместимым со следующими версиями Windows: Windows 2000/XP/2003/Vista.

Драйвер-фильтр файловой системы

Драйвер-фильтр файловой системы – дополнительный драйвер, который добавляет функции или изменяет поведение файловой системы.

Драйвер-фильтр файловой системы может фильтровать операции ввода-вывода для одной или более файловых систем или томов. В зависимости от природы драйвера-фильтра может журналировать, наблюдать, изменять, или даже прерывать файловые операции. Типичными приложениями, использующими драйвер-фильтр файловой системы, являются антивирусы, программы шифрования, иерархические системы управления дисковым пространством.

Нами был разработан драйвер для мониторинга дисковой активности и сбора статистики.