

P13-2003-11

А. С. Кирилов, В. Е. Юдин

**РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ РЕАЛЬНОГО
ВРЕМЕНИ ДЛЯ УПРАВЛЕНИЯ ЭКСПЕРИМЕНТОМ
В СРЕДЕ MS WINDOWS**

Введение

Системы управления спектрометрами на основе промышленных компьютеров в стандарте VME используются в ЛНФ ОИЯИ с 1995 года. В настоящее время такими системами оборудовано большинство спектрометров. Типичная конфигурация включает VME - компьютер фирмы "Eltec" (процессорный блок E17 на основе процессора 68040, память 16 - 64 Мб, диск 2 - 6 Гб, сетевой контроллер 10 Мб/сек), операционную систему OS/9, программный комплекс SONIX [1]. Хотя на ряде спектрометров (СКАТ, НСВР и др.) эксплуатация этих систем успешно продолжается и в настоящее время, однако моральное старение компьютеров и медленное развитие операционной системы OS/9 вызывают необходимость модернизации существующих систем. Особенно это актуально для спектрометров с позиционно-чувствительными детекторами, где поток данных на порядок выше. К сожалению, замена процессорных блоков VME на современные аналоги требует также обновления и средств разработки программного обеспечения, что в совокупности очень дорого.

Подход, который принят в ряде институтов (ILL, ESRF и др.), позволяет получить новое качество и развить более дешевое направление для будущей модернизации. Основная идея решения заключается в замене процессорного блока VME на внешний компьютер (PC), подсоединенный к крейту VME с помощью VME-PCI адаптера. После такой операции крейт VME из самостоятельного компьютера превращается в сложный контроллер, управление которым полностью возложено на внешний компьютер. Данный подход позволяет сохранить уже созданные аппаратные блоки в формате VME, хотя и требует замены программного обеспечения.

Наиболее естественным претендентом на роль внешнего компьютера является персональный компьютер с операционной системой Windows [2].

При выборе схемы построения программного обеспечения на новой платформе будут сохранены решения, хорошо зарекомендовавшие себя в старой системе. Одним из таких решений является применение базы данных реального времени Varman [3] для организации межмодульного взаимодействия (см. рис.1). Эта база используется и для хранения состояния системы, и для оповещения своих клиентов в необходимых случаях об изменениях этой информации.

Реализация базы данных реального времени для MS Windows получила название WinVarman.

Программа написана на языке C++ [4]. Использовался пакет MS Visual C++ 6.0 и библиотека MFC [5].

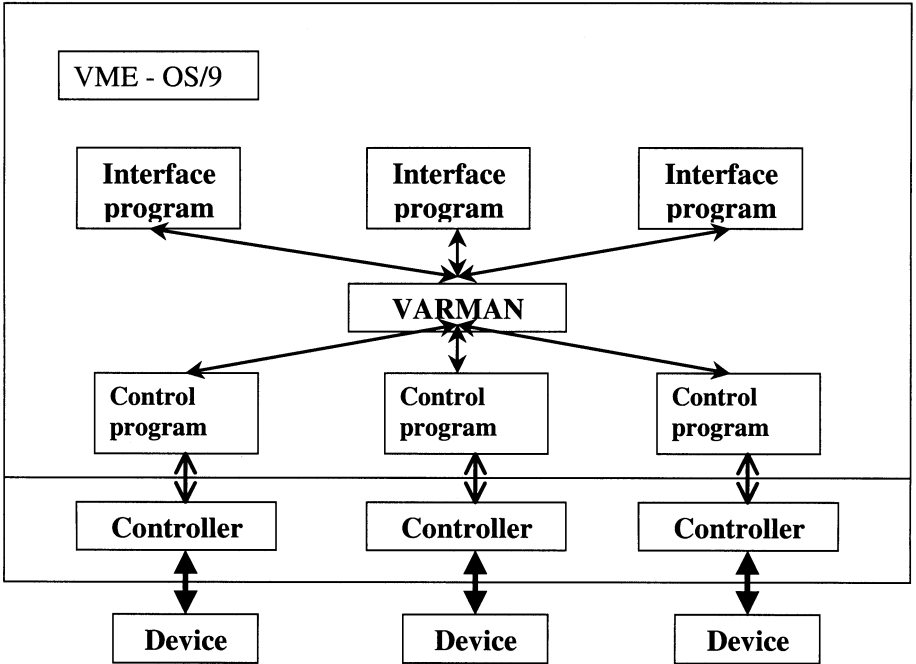


Рис. 1. Архитектура программного комплекса

Требования к WinVarman

Процессы на локальном компьютере должны иметь доступ к общим данным. Каждый блок данных идентифицируется по уникальному имени. Имя - это текстовая ansi-строка с терминальным нулем. Когда блок данных (далее переменная) отвечает определенным условиям (далее интересу), об этом оповещаются заинтересованные клиентские процессы.

Система должна быть надежной и целостной, что подразумевает:

- каждый клиентский запрос оставляет базу данных в корректном состоянии (для этого запрос рассматривается как транзакция – если действие корректное, оно выполняется, если нет - ничего не происходит);
- клиентские запросы не конфликтуют между собой (для решения этой проблемы используются механизмы межпроцессной синхронизации, и запросы выполняются атомарно);
- некорректные работа и завершение клиентского процесса не должны нарушать правильности работы базы данных (следовательно, данные не должны находиться в адресном пространстве клиента, а их копии должны получаться по запросу, что приводит к архитектуре клиент-сервер, и во время ожидания ответа клиента сервер ожидает и завершения работы клиентского процесса).

Передача данных между процессами должна производиться с максимальной скоростью, но не в ущерб описанным выше требованиям. Скорость передачи данных зависит от механизма межпроцессного взаимодействия (далее IPC – interprocess communications). Тип механизма IPC очень важен, поэтому рассмотрим его более подробно.

Выбор механизма IPC

Отметим, что сложность передачи данных между процессами в MS Windows заключается в том, что каждый процесс работает в своем независимом виртуальном адресном пространстве, и адресные пространства других процессов для него напрямую недоступны (исключение MS Windows 9x). Для решения этой задачи существуют различные механизмы IPC. В табл. 1. [6] приведены основные механизмы IPC. Нас интересуют механизмы, которые отвечают следующим требованиям:

- максимальная скорость передачи данных;
- двусторонняя передача данных;
- надежность;
- передача любых (ограниченных) объемов данных;
- удобство использования.

Этим требованиям лучше всего отвечают, на наш взгляд, файлы, отображаемые в память (далее MMF – memory mapped file). Механизм MMF является фундаментальным в MS Windows и все остальные механизмы IPC фактически построены на основе MMF. Таким образом при использовании данного механизма мы получаем высокую эффективность работы.

| Механизм IPC | Комментарии |
|--|--|
| Динамический обмен данными (Dynamic Data Exchange - DDE) | Устарел и труден в реализации. Рекомендуется использовать лишь в случаях крайней необходимости. |
| Буфер обмена | Пересылка данных обычно выполняется с участием пользователя. |
| OLE 2.0 | Имеются встроенные функции для пересылки данных через границы процессов. Излишне сложен для простого обмена данными. |
| Сообщения WM_USER | Они работают - но не допускайте ошибки при передаче указателей. |
| Сообщения WM_COPYDATA | Лучший способ пересылки блока данных из одной программы в другую. |
| Анонимные каналы (Anonymous pipes) | Полезны для организации прямой связи между двумя процессами на одном ПК. |
| Именованные каналы (Named pipes) | Полезны для организации прямой связи между двумя процессами на одном ПК или в сети. |
| Почтовые ячейки (mailslots) | Полезны для организации связи одного процесса со многими на одном ПК или в сети. |
| Гнезда (sockets) | Полезны для организации пересылки данных как в Windows-программы, так и в прочие программы, функционирующие на одном ПК, в сети или в интрасети. |
| События, взаимные исключения (mutexes) и семафоры | Только для простого взаимодействия. Пересылка данных невозможна. |
| Вызов удаленных процедур (remote procedure calls - RPC) | Слишком сложен, чтобы использовать его для простых пересылок данных. |
| Разделяемая память | Непросто выделить вне DLL. |
| Файлы отображаемой памяти | Обеспечивают одновременный доступ к объектам файла отображения из нескольких процессов. |

Табл. 1. Механизмы IPC системы Windows NT

Схема взаимодействия базы данных с клиентами

База данных реализована как отдельная программа. Функции для доступа к базе данных находятся в библиотеке динамической компоновки (далее dll – dynamic link library), которую используют клиенты. MMF используется только для передачи данных.

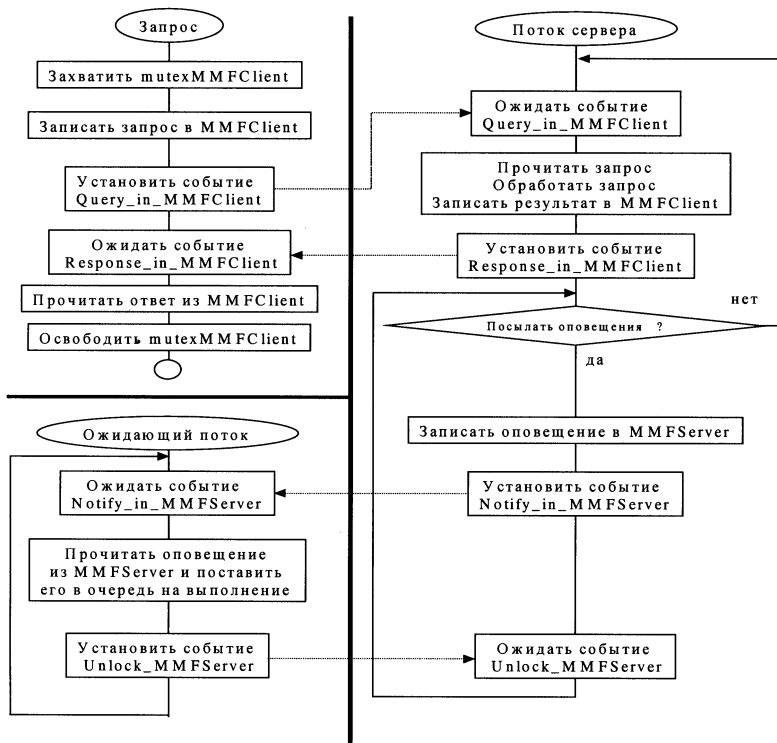


Рис. 2. Схема взаимодействия базы данных с клиентами

Синхронизация выполняется с помощью объектов ядра MS Windows. Может быть запущена только одна копия сервера базы данных. Если хотя бы один клиент подключен к базе данных, то процесс сервера базы данных не может

быть завершен. Схема взаимодействия базы данных с клиентами приведена на рис. 2. Чтобы выполнить запрос к серверу клиенты конкурируют за мьютекс (от mutex – mutual exclusion). Мьютекс был выбран потому, что в MS Windows он освобождается в случае, если поток, захвативший его, завершается. В этом случае некорректно завершенный процесс клиента не может заблокировать доступ к серверу. Все объекты синхронизации типа «событие» (event) являются событиями с автосбросом (autoreset events), т.е. ожидающая функция сбрасывает событие. В точках синхронизации на стороне сервера ожидается и завершение процесса клиента (на рисунке не показано). Для получения извещения от сервера на стороне клиента работает специальный поток.

Клиентские функции

Все клиентские функции возвращают результат выполнения (DWORD). Их можно разбить на следующие группы:

1. VRMN_Connect, VRMN_Disconnect – соединение с сервером и отключение от него;
2. VRMN_SaveState, VRMN_LoadState – запись/чтение состояния базы данных;
3. VRMN_CreateVariable – создание переменной;
4. VRMN_IsExistVariable, VRMN_TypeControlVariable – проверка существования и типа переменной;
5. VRMN_SetInterest – установка интереса к переменной;
6. VRMN_SetVariable, VRMN_GetVariable – установка/чтение значения переменной;

Клиент может установить к переменным следующие типы интересов:

- переменная равна константе;
- переменная не равна константе;
- переменная меньше константы;
- переменная меньше или равна константе;
- переменная больше константы;
- переменная больше или равна константе;
- переменная попадает в диапазон (константа 1, константа 2);
- переменная не попадает в диапазон [константа 1, константа 2];
- переменная изменилась;
- переменная была установлена.

В функцию VRMN_SetInterest передается адрес функции, которая будет вызвана при срабатывании интереса.

Для упрощения работы с указанными функциями были написаны вспомогательные функции на C и классы на C++.

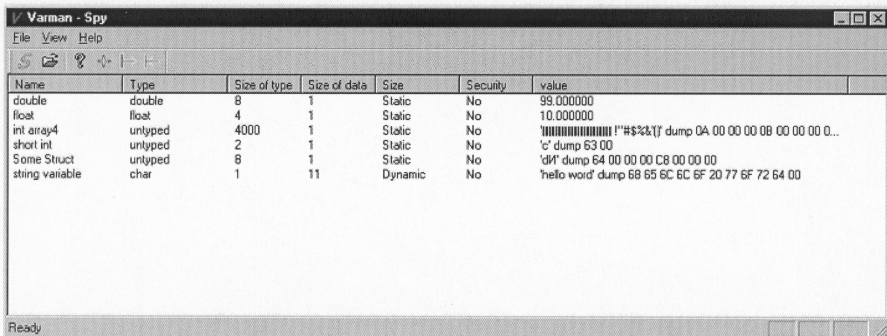


Рис. 3. Интерфейс программы VarmanSpy

Заключение

Для отслеживания состояния сервера и выполнения сервисных операций написана программа VarmanSpy (рис. 3).

Программа прошла практическую проверку в составе системы альтернативного управления спектрометром НЕРА-ПП в течение трех последних циклов реактора в 2002 году и показала высокую устойчивость.

Выражаем благодарность И. М. Саламатину за полезные консультации.

Литература

1. http://nfdfn.jinr.ru/~kirilov/Sonix/sonix_index.htm
2. Дж. Рихтер, Windows для профессионалов: создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows/Пер. с англ. – 4-е изд. – СПб: Питер; М.: Издательско-торговый дом «Русская Редакция», 2001.
3. Proceedings of DANEF'97, pp.288-294, JINR, E10-97-202, Dubna, 1997.
4. Б. Страуструп, Язык программирования C++, спец. изд./ Пер. с англ. – М; СПб.: «Издательство БИНОМ» - «Невский Диалект», 2001.
5. А. Мешков, Ю. Тихомиров, Visual C++ и MFC; Пер. с англ. 2-е изд. перераб. и доп. – СПб.: БХВ-Перербург, 2001.
6. Джефф Просис, Взаимодействие процессов в системе Windows NT / PC Magazine June 24, 1997.

Получено 23 января 2003 г.

Кирилов А. С., Юдин В. Е.

P13-2003-11

Реализация базы данных реального времени
для управления экспериментом в среде MS Windows

Рассматривается реализация базы данных реального времени для комплекса программ управления экспериментом в среде MS Windows на установках реактора ИБР-2. Описаны предъявленные к ней требования, выбор механизма межпроцессорного взаимодействия, схема синхронизации. Особое внимание при реализации базы данных было уделено надежности и быстротедействию.

Работа выполнена в Лаборатории нейтронной физики им. И. М. Франка ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна, 2003

Перевод авторов

Kirilov A. S., Yudin V. E.

P13-2003-11

The Implementation of Real-Time Database for Control
of the Experiment in the MS Windows Environment

The implementation of real-time database for control system on experimental facilities of the IBR-2 reactor in the MS Windows environment is considered. The main demands, the choice of the type of interprocess communication and the synchronization diagram are described. Reliability and processing speed of the program are in the center of attention.

The investigation has been performed at the Frank Laboratory of Neutron Physics, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna, 2003

Редактор *М. И. Зарубина*
Макет *Н. А. Киселевой*

Подписано в печать 13.03.2003.

Формат 60 × 90/16. Бумага офсетная. Печать офсетная.

Усл. печ. л. 0,43. Уч.-изд. л. 0,63. Тираж 305 экз. Заказ № 53807.

Издательский отдел Объединенного института ядерных исследований
141980, г. Дубна, Московская обл., ул. Жолио-Кюри, 6.

E-mail: publish@pds.jinr.ru

www.jinr.ru/publish/