



СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

95-462

Д13-95-462

А.С.Кирилов, Й.Хайнитц

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ
НАКОПЛЕНИЯ, УПРАВЛЕНИЯ И КОНТРОЛЯ
НЕЙТРОННОГО СПЕКТРОМЕТРА
ВЫСОКОГО РАЗРЕШЕНИЯ В СТАНДАРТЕ VME

1995

1. Введение

Ряд лет в Лаборатории нейтронной физики ОИЯИ ведутся текстурные и структурные исследования на нейтронном спектрометре высокого разрешения (НСВР), установленном на седьмом канале импульсного реактора ИБР-2. Система регистрации установки НСВР состоит из семи гелиевых детекторов. Для вращения образца в нейтронном пучке используется двухосный текстурный гониометр. Стандартный режим измерения текстуры предполагает снятие 700 спектров при 100 различных положениях образца.

Система накопления, управления и контроля спектрометра в настоящее время реализована на основе модульного компьютера в стандарте VME. На рис.1 представлена блок-схема этой системы. Детальное описание электронной части системы приведено в /1/ и /2/.

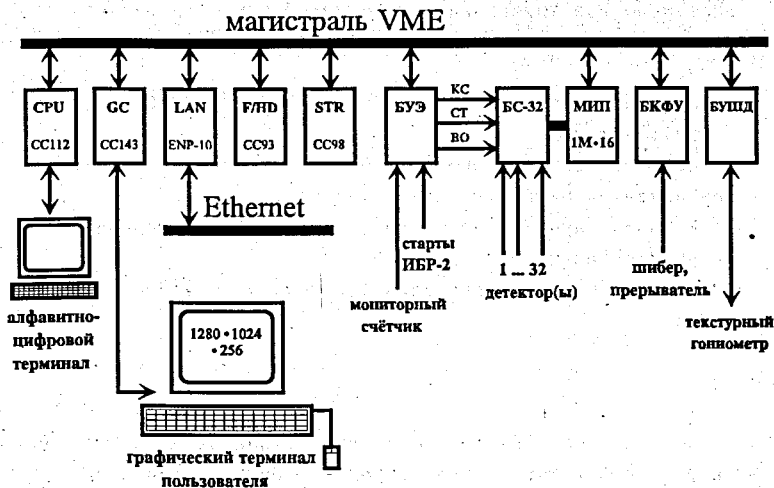


Рис. 1. Блок-схема системы накопления, управления и контроля спектрометра НСВР.

Данная публикация является продолжением работы /2/ и, как следует из названия, посвящена более детальному описанию принципов построения программного обеспечения установки НСВР (далее ПО НСВР), основных моментов его реализации и проблем, встретившихся при этом.

Хотелось бы отметить следующее. Мы создавали конкретную систему исходя из состава оборудования (см. /2/), имеющегося в нашем распоряжении, и программного обеспечения для него. Например, мы имели графический контроллер СС143, цветной монитор и дистрибутив системы X Window, адаптированный для OS-9, но не обладали рабочей станцией, не имели OSF Motif и т.д.

Нашим начальным ориентиром, естественно, была локальная (не сетевая) версия системы, предназначенная для работы непосредственно из экспериментального зала. Однако с самого начала мы искали решение, позволяющее без больших переделок превратить такую систему в распределенную. Причем пользователем особо отмечалось требование сохранения работоспособности системы при отказе или отключении сети.

2. Общие требования

Общие требования к ПО НСВР совпадают с изложенными в работе /3/, а именно:

- непосредственное управление установкой через VME-крейт из экспериментального зала;
- дистанционное управление через локальную сеть лаборатории;
- возможность проведения эксперимента при отказе или отключении сети;
- организация современного многооконного графического интерфейса с пользователем;
- возможность визуального контроля накапливаемых спектров;
- визуализация параметров эксперимента;
- возможность простой адаптации системы при расширении состава экспериментального оборудования;
- легкость изменения программы (процедуры проведения) эксперимента.

3. Общие идеи

При отсутствии "своей" рабочей станции функции основного компьютера, проводящего эксперимент, были возложены на сам VME-компьютер. Многопроцессная операционная система OS-9 этого компьютера позволяет представить всю создаваемую систему как набор

взаимодействующих самостоятельных задач, что позволяет разрабатывать, отлаживать и вводить в эксплуатацию систему поэтапно. Этот подход также позволяет легко наращивать систему при расширении состава экспериментального оборудования, например, при добавлении печек, камер высокого давления и проч., дополняя систему новыми задачами.

Поскольку все задачи должны работать в едином комплексе, важен выбор основы для реализации их "общения". Штатные возможности межзадачного взаимодействия OS-9 хороши только для работы в рамках VME-компьютера и совершенно лишены сетевых средств. Пакет ISP, которым мы располагали, в принципе, использовать возможно, но, на наш взгляд, программирование в этом случае было бы более объемным и сложным.

Идея использования ПО X11/OS-9 /4/ не только в качестве средства организации многооконного интерфейса, но и в качестве основы для реализации протокола межзадачного общения оказалась плодотворной. Если реализовать отдельные задачи в виде X-клиентов, а взаимодействие между ними организовать средствами системы X Window, то система получается сетевой без дополнительных издержек. Дело в том, что при соблюдении некоторых правил, о которых мы сообщим ниже, X Window обеспечивает правильную коммутацию любого X-клиента с любого компьютера сети с любым X-сервером (рабочей станцией, X-терминалом и т.д.) без изменений в программе этого клиента. Причем клиент может работать и в локальном режиме на своем сервере, даже если отключить от этого сервера сеть.

Таким образом, реализация отдельных задач в виде X-клиентов, и организация взаимодействия задач средствами X Window, позволяют ликвидировать различие между локальной и сетевой версиями системы.

Наиболее простой и гибкой формой задания процедуры проведения эксперимента является форма текстового файла. В этом случае пользователь может не только легко создать такую программу с помощью обычного редактора, но даже выполнить корректировку (например, для повтора отдельных измерений и т.д.) непосредственно в ходе самого эксперимента. Для интерпретации этой процедуры необходима специальная программа, управляющая и синхронизирующая работу других программ, представляющих отдельные части системы.

Наконец, мы полагаем, что следует максимально использовать штатные возможности операционных систем. Например, доступ пользователя к тем или иным программам, каталогам, файлам может регламентироваться средствами файловых систем, которые и следует применять для этих целей.

4. Классификация взаимодействия задач

Все задачи, составляющие ПО НСВР, можно разделить на три группы.

Первую группу составляют задачи, которые реально участвуют в процессе управления экспериментом. К ним надо отнести задачи мониторинга пучка, интерпретатора программы эксперимента, управления гониометром и экспозицией. Поскольку все эти задачи управляют экспериментом, они должны быть защищены от повторного вызова. Задачи управления гониометром и экспозицией должны, кроме того, "уметь" работать как самостоятельно, так и под общим управлением задачи, интерпретирующей программу эксперимента.

Вторую группу образуют задачи, которые в процессе управления экспериментом прямо не участвуют, но информируют о его важных моментах. К ним мы отнесли задачу визуализации спектров и информационную задачу. Количество копий этих задач, одновременно работающих в системе, не ограничено, поскольку они не мешают ни процессу измерения, ни друг другу.

К третьей группе отнесены специальные задачи, а именно драйверы электронных блоков, составленных по методике /5/, а также (как мы их назвали) X-драйверы. Подробнее об этом сообщается в разделе 6.2.

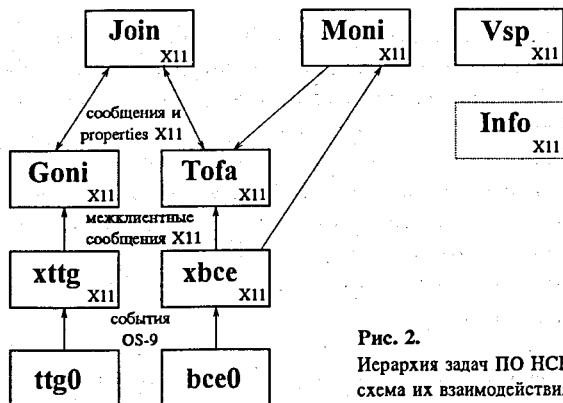


Рис. 2. Иерархия задач ПО НСВР и схема их взаимодействия

На рисунке 2 представлены задачи ПО НСВР и схема их взаимодействия. Знаком X11 в правом углу помечены задачи, являющиеся клиентами X11. Задача Info, обозначенная штриховой рамкой, пока не

реализована, а задача Moni реализована в упрощенном варианте. На рисунке также обозначены средства X11 и OS-9, использованные для организации межадачного обмена.

5. Назначение отдельных задач

Примерный, к сожалению, черно-белый общий вид окон ПО НСВР представлен на рис. 3. Эта и остальные приведенные ниже иллюстрации получены как моментальный снимок с экрана Sparc-station. Рассмотрим основные особенности каждой задачи.

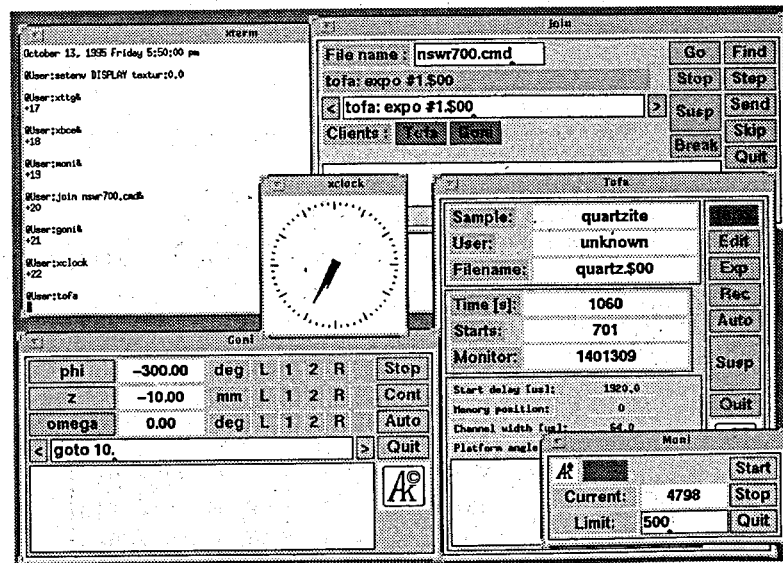


Рис. 3. Общий вид окон комплекса ПО НСВР.

Задача Join (рис. 4) предназначена для интерпретации программы эксперимента, заданного в виде текстового файла. Каждая строка этого файла интерпретируется либо как комментарий, либо как команда. Последняя состоит из имени клиента, оканчивающегося двоеточием, и той строки, которую ему необходимо послать. В задаче реализован широкий

набор операций с командами, включающий непрерывное или пошаговое выполнение программы эксперимента, пропуск, редактирование и вставку команд, останов после выполнения текущей команды, аварийный останов и проч.

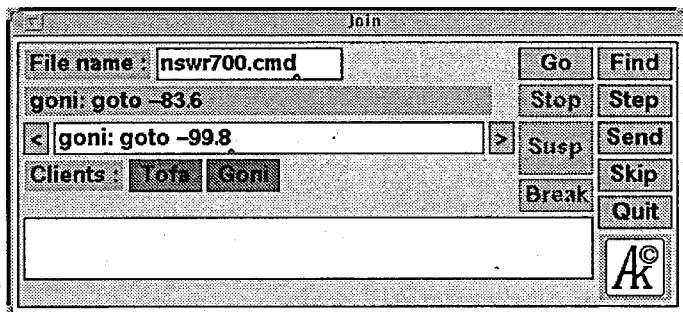


Рис. 4. Окно задачи Join

задача Goni (рис. 5) предназначена для управления гониометром посредством блока БУШД /2/.

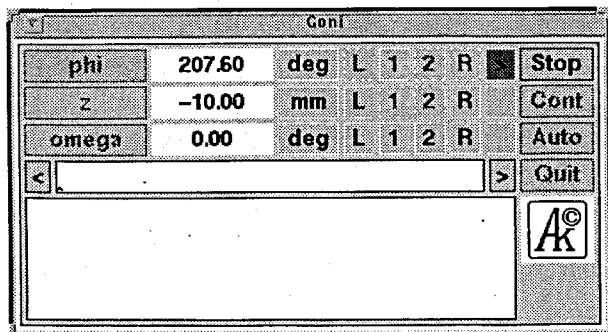


Рис. 5. Окно задачи Goni

Управление двигателями гониометра ведется последовательно. Перемещение двигателя задается в единицах (град, мм), указанных для этого двигателя. Новая позиция задается либо в относительных координатах, либо в абсолютных, если таковая система была предварительно определена. Имеется возможность контроля за невыходом из пределов заданного пользователем диапазона. Задача информирует пользователя о текущей позиции, состоянии контактных переключателей и направлении движения двигателя. В настоящее время число двигателей и их параметры задаются встроенной таблицей. В последующих версиях предполагается задавать их в файле конфигурации. В режиме "Auto" задача переключается под управление задачи Join. В приложении приведен список команд для управления гониометром.

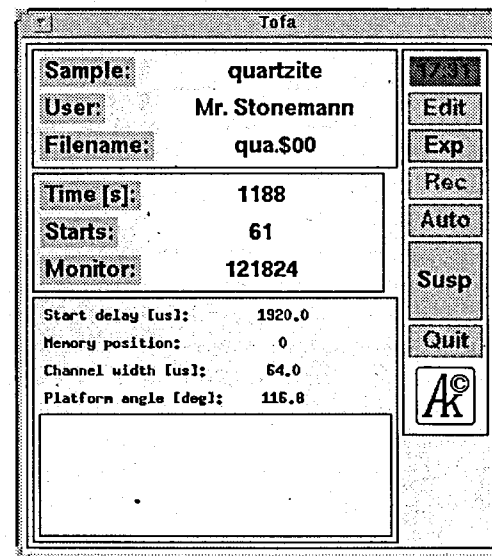


Рис. 6. Окно задачи Tofa

Задача Tofa (рис 6.) служит для управления экспозицией с помощью первого комплекта счетчиков блока БУЭ /2/ и сохранением спектров, накопленных в инкрементной памяти МИП /6/. Экспозиция задается по

любому из трех возможных параметров: времени, числу циклов реактора, значению мониторингового счетчика. Имеется возможность задать имя пользователя, название образца и имена файлов для сохранения спектров. В режиме "Auto" задача переключается под управление задачи Join и выполняет команды, заданные в файле программы эксперимента. В настоящее время этот набор состоит из единственной команды, имеющей следующий синтаксис:

exp0 <string>.

где в <string> с помощью специальных символов "#" и "\$" из имени файла, заданного пользователем, и аббревиатур детекторов формируются имена файлов, под которыми спектры будут сохранены на диске. Кроме того, в режиме "Auto" задача может протоколировать свою работу. Помимо Join Tofa взаимодействует также с задачей Moni.

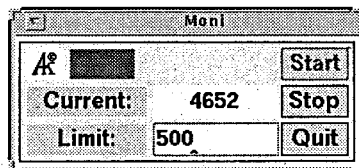


Рис. 7. Окно задачи Moni

Задача мониторинга пучка Moni должна следить за соблюдением условий, необходимых для проведения дифракционного эксперимента на спектрометре, а именно наличием и частотой стартов реактора, фазировкой прерывателя, открытием шибера пучка, интенсивностью потока нейтронов и т.д., а также вести соответствующий протокол. Задача разрешает или запрещает включение экспозиции в задаче Tofa. Для контроля указанных выше параметров используется второй набор счетчиков блока БУЭ и входные регистры блока контроля за состоянием физической установки (БКФУ). На рис. 7 представлено окно упрощенного варианта задачи, который реализован в настоящее время. В этом варианте контролируется интенсивность потока нейтронов. Если интенсивность меньше заданного пользователем значения порога, Moni сбрасывает флаг разрешения включения экспозиции и информирует об этом задачу Tofa. Флаг восстанавливается вновь, когда интенсивность вновь превысит значение порога.

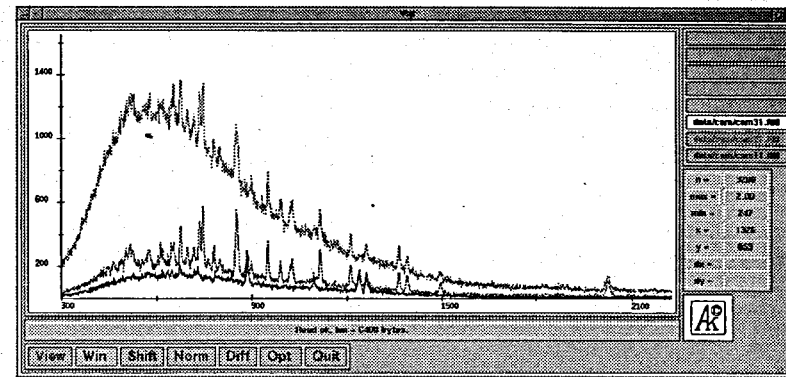


Рис. 8. Пример изображения спектров в задаче Vsp

Задача Vsp (рис. 8.) предназначена для одновременной визуализации до восьми спектров либо накапливаемых в инкрементной памяти, либо хранящихся на диске полученных ранее спектров. Она позволяет в ходе эксперимента следить за процессом накопления спектров, выполняя при этом такие операции, как несколько видов масштабирования, сравнение, нормировка, вычитание фона.

6. "Технические" вопросы реализации

Среди особенностей реализации особо отметим следующие:

- выбор средств программирования в системе X Window;
- выбор схемы обслуживания прерываний и управления X-клиентами;
- выбор схемы межклиентного (междузадачного) обмена.

Остановимся подробнее на каждой из особенностей.

6.1 Выбор средств программирования в системе X Window

Мы располагали программным обеспечением X Window System X11/OS-9 v 1.1 фирмы CompControl International B.V. (Голландия), включающим как серверную, так и клиентскую части, в том числе библиотеку виджетов (widgets) Athena. При отсутствии удобного коммерческого набора виджетов, таких как OSF Motif, мы опробовали ряд свободно распростра-

няемых наборов виджетов (Xw, Free Widget Foundation) и остановились на библиотеке Libsx, полученной по сети Usenet и адаптированной для OS-9 нами самостоятельно. Автором этой библиотеки является Dominic Giampaolo (dbg@sgi.com). Он поясняет:

"Libsx - "простая X-библиотека" - набор программ, базирующихся на наборе виджетов Athena. Целью составления библиотеки было желание упростить написание X-программ. Для этого libsx скрывает массу мелких или "уродливых" деталей, присутствующих обычно в X-программах, и заполняет пробелы, имеющиеся в наборе Athena (такие как виджет для рисования). Libsx стремится упростить общий случай до вызова единственной функции с небольшим числом параметров."

Более того Libsx не препятствует прямому использованию процедур X Window Toolkit или Xlib, чем мы и воспользовались в ряде случаев, но в целом ее применение на, наш взгляд, себя оправдало.

6.2 Схема обслуживания прерываний

В работе /7/ описан сравнительно удобный в рамках OS-9 способ подключения процедуры обслуживания прерывания. Согласно этому способу процедура обслуживания прерывания, размещенная в упрощенном драйвере, информирует основную программу о прерывании с помощью событий (events) OS-9. В работе /5/ предложены программные средства, реализующие этот способ. Однако для X-клиентов эта схема нуждается в дополнении. Необходимо "перевести" с "языка" OS-9 на "язык" X-протокола, т.е. превратить событие OS-9 в X-событие. Как известно, программа каждого X-клиента после начального описания переходит в т.н. основной цикл - цикл обработки X-событий. Ввести обработку событий OS-9 в этот цикл можно либо в форме рабочей процедуры, либо заменив стандартную подпрограмму основного цикла на свою с необходимыми дополнениями, а именно опросом состояния искомого события. К сожалению, и в том и в другом случае X-клиент начинает в значительной степени впустую занимать все доступное ему процессорное время, мешая другим клиентам. "Усыпить" клиента до появления необходимого события (прерывания) нельзя, поскольку в этом случае нарушится нормальная обработка X-событий для клиента.

Наше решение (смотри схему на рис. 2) заключается в составлении для обработки каждого события OS-9 (прерывания) отдельной программы, которую мы назвали X-драйвером. Эти программы являются X-клиентами и поэтому в рамках X-протокола могут вести нормальный диалог с другими X-клиентами, в частности, посылать и принимать X-сообщения. Однако в них отсутствует процедура основного цикла, которая заменена процедурой

ожидания события (прерывания) в "спящем" режиме. Во время ожидания события X-драйвер "спит" и не занимает процессорное время. Кроме того, в отличие от других клиентов X-драйвер не имеет окна, что, в частности, дает экономию оперативной памяти. Если размер обычного клиента составляет несколько сот кбайт, то размер X-драйвера - около 30 кбайт.

Двухуровневая схема обработки прерываний позволяет удобно разделить работу между драйверами: действия, требующие быстрой реакции, обслуживаются нормальным драйвером, а остальное - X-драйвером либо самим клиентом. Так, например, в текущей версии после истечения времени экспозиции по прерыванию управляющий бит разрешения экспозиции в регистре управления блока БУЭ сбрасывается процедурой обслуживания прерывания драйвера все0, а остальные действия выполняются задачей Tofa по получении соответствующего сообщения от хвсе.

6.3 Схема межклиентного обмена

В нашей программе на межклиентный обмен возложены следующие функции:

- поддержка протокола между задачей Join и подчиненными клиентами;
- информирование клиентов о прерываниях;
- информирование клиентов об условиях проведения эксперимента, определяемых задачей Moni;
- формирование и поддержка набора данных, содержащих сведения о значениях ряда параметров, заданных в конфигурационном файле, а также отражающих текущее состояние процесса эксперимента для информационных задач.

Как отмечалось выше, все это реализуется только средствами X11, а именно с помощью атомов (atoms), properties и межклиентных сообщений (client messages). В системе X все эти средства связаны с конкретными серверами (displays) и окнами (windows). Поскольку мы предполагаем, что наши клиенты могут запускаться не только на VME-компьютере, но на других компьютерах сети ЛНФ, для того чтобы процесс обмена информацией между клиентами шел корректно во всех случаях, мы предлагаем следующее решение.

Вводится понятие базового X-сервера спектрометра (установки), на котором регистрируются все атомы и все properties, участвующие в обмене. Каждый клиент в начале своей работы устанавливает связь с базовым сервером и регистрирует там свой сервер и окно. Там же он ищет "координаты" нужных ему для отправки сообщений клиентов.

Понятно, что во время работы спектрометра базовый сервер должен быть всегда включен и доступен другим компьютерам сети. Естественно, что в нашем случае в качестве базового был выбран сам компьютер VME. Важно, что при этом выборе работоспособность программы сохраняется даже при отключении компьютера от сети.

7. Программные трудности и способы их преодоления

При работе с X-пакетом фирмы ComrControl мы столкнулись с рядом ошибок, а именно:

1). При относительно быстрой генерации графических изображений, состоящих из большого числа векторов (например, гистограмм), появляются сбои в работе X-конвейера. Для уменьшения скорости генерации изображения мы отказались от использования сопроцессора в задаче Vsp.

2). Мы обычно наблюдали досадные изменения последовательности выполнения операций в X-конвейере: Так, вместо стирания подокна и последующего перерисования изображения в основном окне, как это определено в программе, эти операции выполняются в обратном порядке.

3). Виджет Text набора Athena содержит многочисленные ошибки.

4). X-клиенты часто некорректно используют систему динамического выделения памяти OS-9. Некоторые виджеты постоянно занимают новый блок памяти при изменении своих ресурсов. Это приводит к аварийному окончанию работы клиента и даже к отказу операционной системы.

5). Иногда клиенты не получают вовремя сообщений по таймауту, что выглядит так, как будто бы они "заснули".

Наша попытка обратиться в фирму за помощью оказалась безрезультатной, поскольку последняя фактически отказалась. Пришлось обходиться своими силами. Из всех ошибок, перечисленных выше, самой существенной являются сбои в работе системы динамического выделения памяти, поскольку они приводят к постоянному сокращению объема свободной памяти и вплоть до разрушения системных структур. Время корректной работы системы было мало и непрогнозируемо. Мы провели исследование и выяснили какие виджеты и действия способствуют этому процессу. Затем задачи были исправлены так, чтобы минимизировать вредный эффект. В настоящее время программа стабильно работает полное время измерения одного образца.

8. Заключение

Данная работа была выполнена в период с весны 1994 г. по март 1995 г. С 27 марта 1995 г. VME-система с первой версией ПО НСВР эксплуатируется на спектрометре вместо прежней системы, базирующейся на КАМАКЕ и РС АТ. Эта версия не позволяет пока полностью распределенного управления, но необходимые для этого изменения практически опробованы.

Отметим еще раз основные черты, присущие нашему подходу:

- представление всего комплекса как совокупности X-клиентов, реализованных на основе ПО X11/OS-9;
- введение понятия базового сервера спектрометра (установки) и использование X-сервера ПО X11/OS-9 на VME-компьютере в качестве него;
- единство локальной и сетевой версий ПО НСВР;
- введение понятия X-драйвера и построение схемы обслуживания прерывания с его использованием;
- использование набора виджетов Athena и библиотеки Libsx для создания клиентов.

9. Признательность

Авторы считают своим долгом поблагодарить коллег-электронщиков А.П.Сиротина, М.Л.Коробченко и В.Е.Резаева за плодотворное сотрудничество; К.Уллемайера и Д.И.Николаева за терпение и полезные замечания; Д.Кемпа, Й.Петерсена, М.Меркеля (все ЦЕРН) и Ю.А.Астахова за консультации и помощь в практических вопросах; В.И.Приходько за поддержку работы.

10. Приложение

Список команд для управления шаговыми двигателями (задача Goni)

- <name> - выбор в качестве активного двигателя name;
- move x - перемещение активного двигателя на x единиц относительно текущей позиции;
- goto x - абсолютное перемещение активного двигателя в позицию x;
- setpos x [y z] - переопределение текущей позиции как позиции x; если заданы y и z, то они определяют допустимый диапазон перемещений двигателя;
- getpos - чтение текущих позиций двигателей (и диапазонов) из служебного файла;
- find l|r|1l|1r|2l|2r - поворот двигателя в указанном направлении (вправо или влево) до достижения указанного контакта:
- l - левый концевик;
 - 1 - первый промежуточный контакт;
 - 2 - второй промежуточный контакт;
 - r - правый концевик.

11. Литература

1. CompControl International B.V. System Hardware Support Manual Version 1.0, March 1992.
2. Зем Ен Кен и др. Система накопления, управления и контроля спектрометра НСВР в стандарте VME. ОИЯИ, P13-94-73, Дубна, 1994.
3. В.Е.Новожилов и др. Организация распределенной системы в стандарте VME для автоматизации экспериментов на реакторе ИБР-2.ОИЯИ, P10-94-8, 1994.
4. CompControl international B.V. X window System X11/OS9 User's Manual, VERSION 1.1, March 1991.
5. А.С. Кирилов. Методика организации упрощенного доступа к устройствам в операционной системе OS9 и средства ее реализации. ОИЯИ, P10-94-151, Дубна, 1994.
6. Резаев В.Е. Спектрометрическое накопительное запоминающее устройство в стандарте VME. Труды XVI международного симпозиума по ядерной электронике (Варна, Сентябрь 12-18, 1994), D13-94-491, Дубна, 1995.
7. J.O. Petersen, P. Scarff-Hansen. CAMAC Routines for OS-9 /VMEbus/CBD8210 Systems. CERN/ECP/DS, 1992.

Рукопись поступила в издательский отдел
15 ноября 1995 года.