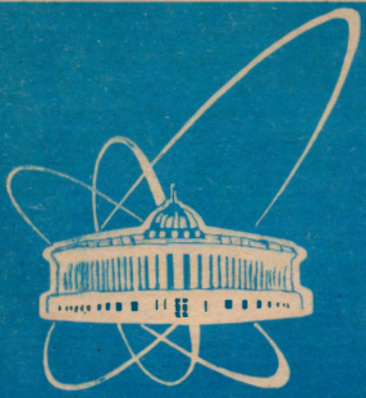

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

Р13-97-161

А.С.Кирилов, Й.Хайнитц*

ИНТЕРПРЕТАЦИЯ ПРОЦЕДУРЫ ЭКСПЕРИМЕНТА В ПРОГРАММНОМ КОМПЛЕКСЕ СИСТЕМ НАКОПЛЕНИЯ, УПРАВЛЕНИЯ И КОНТРОЛЯ СПЕКТРОМЕТРОВ **НСВР** И **СКАТ** (ЗАДАЧА **JOIN**)

*Leica Lithographic Systems, Jena, Germany

1997

Введение

Система накопления, управления и контроля на основе VME-системы/1/ была первоначально разработана для нейтронного спектрометра высокого разрешения (НСВР), а затем с некоторыми необходимыми исправлениями и добавлениями перенесена на спектрометр количественного анализа текстуры (СКАТ)/2/. Процедура проведения эксперимента на обоих спектрометрах состоит из совокупности простых операций: поворот образца, экспонирование, обычно 15 - 30 минут, вновь поворот и т.д. Полное измерение одного образца занимает порядка суток, и поэтому при наличии даже очень удобного интерфейса возникает естественное желание поручить все это управляющему компьютеру. Подобную процедуру проведения эксперимента мы назвали программой эксперимента.

Эта программа задается в виде текстового файла, который пользователь составляет обычным редактором. Одним из первоначальных намерений было поручить эту функцию задаче Tofa/3/, которая управляет экспозицией и записью спектров и потому занимает особое положение в комплексе. Однако затем было принято решение выделить интерпретацию программы эксперимента в отдельноую задачу, которая была названа Join.

Рис.1 иллюстрирует положение задачи Join в комплексе НСВР (СКАТ).

Общие идеи

Фактически, Join является формальным (т.е. не связанным со спецификой отдельных задач-исполнителей) интерфейсом между созданным пользователем текстовым файлом, содержащим последовательность команд для выполнения многошаговой программы эксперимента и задачами, управляющими отдельными узлами спектрометра, которые далее будут называться клиентами Join. Этот формализм позволяет сделать Join независимой не только от специфики отдельных клиентов, но и от специфики самого спектрометра. Для работы с Join клиенты должны быть переведены в автоматический режим, который включается кнопкой "Auto" на каждом клиенте, либо ключом "-auto" при запуске клиента.

Программа выполняется последовательно шаг за шагом.

Каждая строка файла интерпретируется либо как комментарий, если она начинается с точки с запятой, либо как команда. Последняя состоит из имени клиента, оканчивающегося двоеточием, и той строки, которую ему необходимо послать. На рис.2 в качестве примера приведен фрагмент одной из измерительных програм для НСВР.

Join может выполнять программу как в режиме прокрутки, так и в пошаговом режиме. В режиме прокрутки программа последовательно выполнится до конца, если этот процесс не будет остановлен пользователем или появлением ошибки в одной из команд. Выполнение программы также прерывается при достижении строки "#stop".

Все команды могут быть разделены на два типа: прерываемые и непрерываемые.

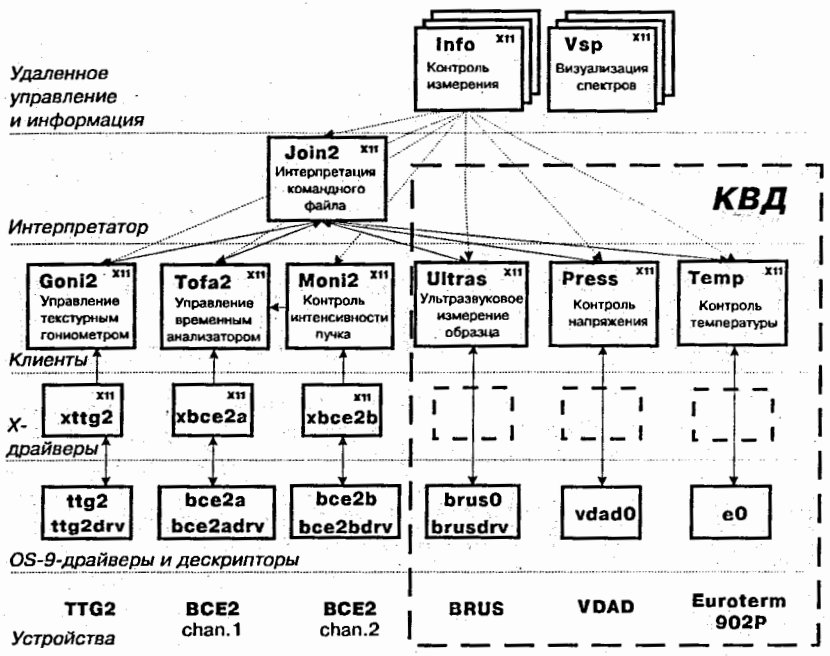

Рис.1. Состав, иерархия и схема связи задач программного комплекса НСВР (СКАТ)

К прерываемым командам относятся те, выполнение которых может быть прервано или приостановлено пользователем. Характерным примером такой команды является команда экспозиции Expo в задаче Tofa/3/. Обработка прерываемых команд выполняется в два этапа: прием и исполнение. После получения команды клиент посылает промежуточное сообщение - подтверждение о приеме, а после выполнения - сообщение с результатом.

Для непрерываемых команд промежуточное сообщение не высылается. Для предотвращения возможной неопределенности при отсутствии или потере ответа от клиента в Join организуется контроль времени ответа.

Состав команд для каждого клиента определяется его предназначением и носит индивидуальный характер, но одна команда является общей для всех клиентов. Это - команда Test, предназначенная для проверки того, находится ли клиент в автоматическом режиме. С помощью набора этих команд, адресуемых по очереди всем клиентам комплекса, пользователь в начале эксперимента может проверить наличие связи и готовность каждого клиента.

Важной функцией Join является ведение общего протокола для всего комплекса. Этот протокол содержит в сжатом виде основную информацию об измерении, источниками которой выступают как клиенты, так и сам интерпретатор. Запись в основной протокол выполняется только интерпретатором. Клиенты могут вести свои "частные" протоколы, например, в целях отладки, если это предусмотрено при их создании.

```
;
;       NSHR measurement procedure
;       File name: nswr700.119.cmd
;       Authors: J.Heinitz & K. Ullemeyer
;       Modified by V.V.Luzin
;
Tofa: TEST
Goni: TEST
Goni: GETPOS
Tofa: user luzin
Tofa: sample quartzite 50ab
Tofa: file ab50.$00
Tofa: time 450
Tofa: OPEN_PROT #.txt
#stop
Goni: OMEGA
Goni: FIND R
Goni: FIND 2L
Goni: SETPOS 0.
Goni: GOTO -83.6
Goni: PHI
Goni: SETPOS 0.0
Goni: GOTO 10.
Goni: FIND 1R
Goni: SETPOS 0.
;
; first turn
Goni: Z
Tofa: EXPO #1.$00
...
Tofa: EXPO #2.$49
```

Рис.2. Фрагмент программы эксперимента

По просьбе пользователей в связи с не вполне надежной работой X-пакета фирмы CompControl/5/ был введен режим автоматического запуска клиентов. Пользователь может составить отдельный командный файл для автоматического перезапуска всего комплекса вместе с X-сервером. После запуска

сервера следует запустить X-драйверы /3/, затем с ключом "-auto" задачи первой группы. Каждая задача сама восстановит свое состояние, сохраненное с предыдущего сеанса работы и переключится в автоматический режим. Затем надо запустить Join также с ключом "-auto". Join самостоятельно выполнит команду Find и продолжит выполнение программы с прерванного места.
Мы надеемся, что с переходом на новый, более надежный X-пакет от Кая Томпсена, необходимость в автоматическом перезапуске комплекса отпадет.

Принципы организации связи

Как отмечалось выше, и сама задача Join и ее клиенты, в свою очередь, являются X-клиентами. Для организации их взаимодействия использованы средства X Window System, а именно, atoms (атомы), properties (в этой работе они по сути своего применения будут называться "почтовыми ящиками") и client messages (межклиентные сообщения). X Window System является принципиально сетевой системой и позволяет без больших издержек организовать работу так, что каждый клиент может быть запущен с произвольной рабочей станции или X-терминала локальной сети без нарушения целостности работы всего комплекса. Для этого достаточно выбрать один из X-серверов сети, на котором регистрируются "координаты" (дисплей, окно) всех клиентов и организовано хранение взаимной "корреспонденции". (Этот сервер мы назвали "базовым"/3/.)
В действительности, для задач, составляющих первую группу /3/, т.е. тех, которые реально участвуют в процессе управления спектрометром, такая гибкость выглядит излишней. Трудно себе представить, что кто-то из пользователей захочет, например, запустить задачу управления экспозицией Tofa с компьютера в одном здании, а задачу управления гониометром Goni - с компьютера из другого.
Практически возможными представляются два варианта: все управляющие задачи запускаются с использованием сервера X11/OS-9, работающего на VME-компьютере спектрометра, либо с использованием X-севера какой-нибудь рабочей станции, если первый вариант невозможен, например, вследствие поломки графического контроллера CC143. И в том, и в другом случае все клиенты запускаются с общего сервера, имя которого программно доступно как значение переменной окружения DISPLAY. Этот сервер автоматически объявляется базовым. Поэтому достаточно регистрировать только окна клиентов. Кстати, при применении процессорного модуля E17 /4/, имеющего встроенный графический контроллер, необходимость применения второго варианта может отпасть совсем.
Для задач второй группы /3/, назначение которых состоит во всестороннем информировании пользователей, регистрация должна выполняться в полном объеме. Для них имя базового сервера задается в конфигурационном файле. Пример последнего дан в приложении.

На HCBP и CKAT в качестве базового используется X-сервер VME-компьютера.
Для регистрации каждый клиент в начале своей работы создает на базовом сервере почтовый ящик, в котором записывает идентификатор своего окна. Этот идентификатор используется Join для общения с данным клиентом. Перед окончанием своей работы клиент обязан стереть свой идентификатор. В свою очередь, Join таким же образом регистрирует свое окно, чтобы быть доступной другим клиентам.
Общение между клиентами и Join производится посредством двух специальных почтовых ящиков и межклиентных сообщений. Один из ящиков используется для передачи команд от Join к клиенту, другой - для возврата сообщений, если таковые имеются.
Выбор идентификатора окна задачи производится непосредственно перед обращением посылкой этой задаче сообщения. Поэтому порядок регистрации (и запуска) клиентов несуществен, кроме случая автоматического перезапуска комплекса. Важно лишь, чтобы к моменту обращения было, к кому обращаться. Пользователь может запустить сначала, например, Goni, выполнить при желании необходимую установку гониометра вручную, затем перевести его в автоматический режим, запустить Join и начать выполнение программы.

Протокол взаимодействия интерпретатора с клиентами

На рис.3 изображена схема состояний Join.
Рассмотрим подробнее протокол общения Join со своими клиентами. Для передачи команды клиенту Join помещает ее в почтовый ящик команд CommandPBox и посылает сообщение BATCH_COMMAND соответствующему клиенту. Если клиент не ответит на это послание в течение определенного промежутка времени, выполнение программы (команды) заканчивается, выдается сообщение об ошибке, и Join переходит в пошаговый режим. Получив сообщение BATCH_COMMAND, клиент анализирует его буфер данных. Как известно, в состав межклиентного сообщения входит буфер данных размером 20 байт, в котором можно разместить произвольную информацию. Если буфер пуст, то клиент рассматривает полученное сообщение как требование выполнить очередную команду. Он извлекает ее из почтового ящика и дешифрирует. Если это непрерываемая команда, то она выполняется, и интерпретатору посылается соответствующий ответ. Он состоит из сообщения BATCH_REPLY с кратким сообщением в буфере. В случае необходимости расширенный ответ передается через другой почтовый ящик ReplyPBox. В первый байт краткого ответа помещается реакция на команду:

'+' - команда принята к исполнению (для прерываемых команд);
'a' - команда выполнена;
'e' - при выполнении команды возникла ошибка;
'u' - послана неверная команда или допущена ошибка в параметрах.

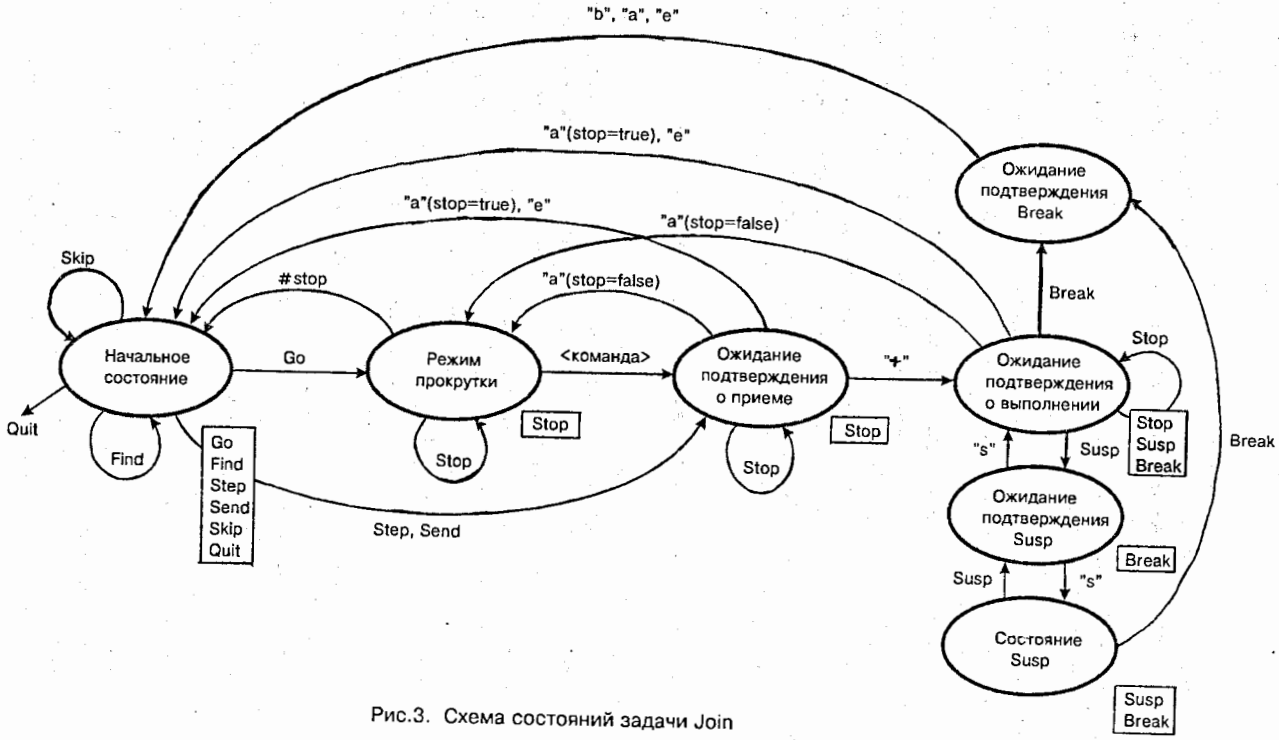

Рис.3. Схема состояний задачи Join

Признаком расширенного ответа служит знак '=' во втором байте. Join извлекает текст расширенного ответа и записывает его в протокол эксперимента. Если это сообщение об ошибке, оно также выводится в поле диагностики.
При выполнении прерываемых команд клиент обычно отвечает дважды. Сначала он посылает сообщение с '+', подтверждая прием команды. Получив это сообщение, Join активизирует кнопки Susp и Break (см. Интерфейс). После завершения выполнения команды посылается сообщение с 'a' или 'e' в соответствии с результатом, и кнопки Susp и Break деактивизируются.
Выполнение команд Susp и Break реализовано иначе, чем для остальных команд в силу чрезвычного характера выполняемых ими действий. Обе команды реализованы по схожим схемам.
По нажатии кнопки Susp Join посылает клиенту сообщение BATCH_COMMAND со строкой "SUSP" в буфере данных и ожидает подтверждения. Join не может самостоятельно приостановить (или прекратить) выполнение текущей команды, поскольку это может сделать только соответствующий клиент. Клиент выполняет (или отклоняет) требуемое действие и отвечает обычным путем через сообщение BATCH_REPLY и при необходимости передать расширенный ответ - через ReplyPBox. Возможны следующие варианты ответов:
"se" - требование отклонено (к моменту его получения выполнение прерываемой команды уже закончилось);
"sa" - команда приостановлена;
"s=" - команда приостановлена и расширенный ответ помещен в ReplyPBox.
Получив подтверждение, Join завершает "оформление" команды (см. "Интерфейс"). Повторное нажатие кнопки Susp означает возобновление приостановленной команды и вызывает повторение всей процедуры с противоположным назначением.
По аналогичной схеме выполняется команда Break.
Заметим, что момент подтверждения запроса на Break или Susp необходим, поскольку обмен сообщениями между Join и клиентами происходит не мгновенно, особенно в случае, когда он выполняется по сети, а не внутри VME-компьютера. Если в период ожидания Susp или Break Join получит сообщение о завершении текущей команды, выполнение программы будет остановлено, как в случае команды Stop.
В заключение отметим одну особенность протокола. Хотя в целом общение Join со своими клиентами не связано со спецификой конкретных команд, есть исключение из этого правила. Это команда Open_prot для Tofa. По традиции, установившейся на НСВР, имя протокольного файла тесно связано с именами файлов, предназначенных для сохранения спектров, которые задаются в задаче Tofa. Получается, что файл с протоколом измерения ведется одной задачей, а имя этого файла должно быть определено в другой. В качестве решения проблемы было предложено следующее. Задача Join вначале своей работы открывает файл с фиксированным именем

join.def.prot. Задача Tofa по команде Open_prot открывает новый файл с нужным именем, записывает (если файл открывается впервые) заголовок протокола, закрывает файл, помещает его имя в почтовый ящик ReplyPBox, а вместо стандартного ответа "a=" помещает в буфер данных сообщения - "a*". Получив ответ, Join извлекает новое имя протокольного файла и далее ведет протокол уже в этом файле.

Интерфейс

На рис.4 представлено окно задачи Join. В левой части окна расположены в основном информационные поля, в правой - кнопки управления процессом выполнения программы эксперимента.
В строке Clients размещен набор кнопок, формируемый в соответствии со списком потенциальных клиентов, который задан в конфигурационном файле. Цвет кнопки отражает текущее отношение к данному клиенту. Если цвет желтый, то клиент еще не зарегистрировался, насыщенный - зарегистрировался. Нажав на кнопку, пользователь может отключить (включить) выполнение команд для данного клиента без изменения самой программы. В этом случае цвет кнопки, соответствующей клиенту, меняется на красный.
Если в результате какой-либо команды произошла ошибка, правее списка клиентов появляется красное поле с надписью "Error". После посылки следующей команды это поле изчезает.
Ниже, справа от метки File name, расположено имя файла программы эксперимента. Для ввода следует перевести курсор мыши на это поле и набрать имя на клавиатуре, обязательно закончив ввод нажатием клавиши "Enter". Окончание ".cmd" в имени файла может быть опущено. По нажатии "Enter" задача ищет файл в текущем каталоге пользователя, открывает его, считывает первую исполняемую команду и выводит ее в поле команды. После этого активизируются кнопки для работы с программой.
Имя файла с программой эксперимента может быть задано также в качестве параметра при запуске Join. В этом случае оно автоматически появится в поле имени.
Если пользователь, работая в пошаговом режиме, захочет вернуться к началу программы, ему достаточно поместить курсор мыши в поле имени файла и нажать "Enter" на клавиатуре.
Ниже, одно над другим расположены два поля, в которых пользователь может видеть (верхнее поле) и редактировать (нижнее поле) текущую команду. Редактирование и вставка команд возможны только в пошаговом режиме. Если пользователь хочет отредактировать текущую команду, он должен перевести курсор мыши на нижнее поле, если оно не пусто, стереть его содержимое и нажать "Enter" на клавиатуре. После этого текущая команда (из верхнего поля) будет скопирована в нижнее поле. Выполнив желаемую редакцию, пользователь вновь должен нажать "Enter", и исправленная или заново набранная команда будет перенесена в верхнее поле. Если

длина команды превышает размер поля, пользователь может сканировать ее влево и вправо с помощью кнопок "<" и ">" соответственно.

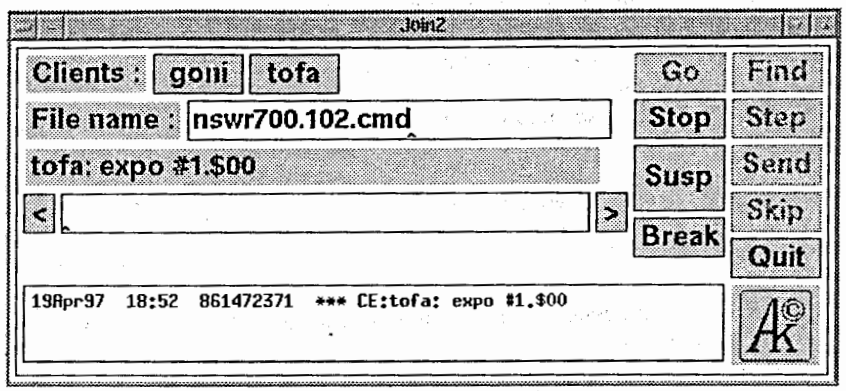

Рис.4. Общий вид окна задачи Join

В самом низу окна находится поле информационных сообщений и диагностики.
Кнопки управления процессом выполнения программы расположены в две колонки в провой части окна Join. В каждый момент времени активизированы только те из них, которые имеют смысл.
Левую колонку составляют кнопки, относящиеся к режиму прокрутки, правую - к пошаговому режиму. Кнопка Go предназначена для запуска программы с текущей команды. Она становится активной, только если введено имя файла с программой. Go активизирует кнопку Stop и деактивизирует кнопки правой колонки. Для прерываемых команд на период исполнения активизируются кнопки Susp и Break.
Кнопка Stop предназначена для остановки выполнения программы после выполнения текущей команды, т.е. перехода из режима прокрутки в пошаговый режим. После первого нажатия она становится красной. Повторное нажатие отменяет остановку.
Кнопка Susp предназначена для приостановки выполнения текущей команды. После получения подтверждения от клиента она становится красной. Время приостановки не лимитировано. Повторное нажатие отменяет требование пользователя на приостановку выполнения команды, и, если за время приостановки не появилось других запрещающих факторов, выполнение команды и программы в целом возобновляется. Например, в случае приостановления пользователем команды Expo задачи Tofa, если во время приостановки был опущен шибер пучка, экспонирование не возобновится, пока шибер не будет поднят вновь, и задача мониторирования пучка Moni не снимет запрет на включение экспозиции.
Кнопка Break служит для немедленного прекращения текущей команды. Прерванная команда остается текущей.
Нажатие Stop, Break, появление конца файла или получение сообщения об ошибке возвращают Join в пошаговый режим и вновь активизируют кнопки правой колонки.

Кнопка Find предназначена для поиска команды, на которой было остановлено выполнение в предыдущем сеансе. В процессе прокрутки Join запоминает позицию, на которой выполнение программы было прервано, и использование Find позволяет вернуться в эту позицию в следующей попытке.
Кнопка Step служит для выполнения текущей команды и перехода к следующей. Если имя файла с программой не было введено, эта кнопка не активна.
Кнопка Send посылает текущую команду для исполнения, не изменяя позиции в командном файле.
Кнопка Skip предназначена для пропуска текущей команды и перехода к следующей. Если имя файла с программой не было введено, эта кнопка не активна.
Кнопка Quit прекращает работу задачи Join.

Заключение

Отметим еще раз основные черты, характеризующие предлагаемый подход:
- удобство управления спектрометром;
- легкость создания программы эксперимента и внесения в нее изменений;
- универсальность задачи Join и по отношению к клиентам, и по отношению к установке (спектрометру);
-. возможность естественного развития системы путем добавления новых клиентов; так если для НСВР были подготовлены Tofa и Goni, то для СКАТа к ним были добавлены Temp, Ultras и Press - для обслуживания подсистемы камер высокого давления.

Приложение. Пример протокольного файла

```
      Measuring protocol of NSHR diffractometer
      =========================================
Period:    March97c2          Procedure:   test.cmd
Sample:    test               User:        kirilov

Time offset:    0 us      Flight path:     0.00 m
Channel width:  64 us     Platform angle:  116.80 deg

 Date     Start     AST          Time[s]    Pulses  Monitor      File
-------------------------------------------------------------------------
 2Apr97  16:42  859995755     232.107    1163  23260001
 2Apr97  16:46  859995988          Measurement was aborted by user.
 2Apr97  17:13  859997632    1200.038    6000  120000001    test1.$00
 2Apr97  17:44  859999455    1200.038    6000  119793631    test1.$01
 2Apr97  17:54  860000060    1200.037    6000  120000001    test1.$02
 2Apr97  18:14  860001270    1200.038    6000  120000001    test1.$03
 2Apr97  18:34  860002479    1200.038    6000  120000001    test1.$04
 2Apr97  18:54  860003689    1200.038    6000  120000001    test1.$05
 2Apr97  19:14  860004898    1200.038    6000  120000001    test1.$06
 2Apr97  19:35  860006107    1200.038    6000  120000001    test1.$07
 2Apr97  19:55  860007316    1200.038    6000  120000001    test1.$08
 2Apr97  20:15  860008525    1200.038    6000  120000001    test1.$09
```


Приложение. Пример конфигурационного файла

```
; Configuration of the time-of-flight analyzer
; ============================================
;
;       Author: Kirilov A.S.  27.03.97
;       NSHR 2.04.97
;
DISPLAY=NSWR_X:0.0              ; base X-server name
;
INTERPRETER= join, JoinWindow, JoinPBox     ; Interpreter parameters
CLIENT= tofa, TofaWindow, TofaPBox          ;  client parameters
CLIENT= goni, GoniWindow, GoniPBox          ;  client parameters
;CLIENT= temp, TempWindow, TempPBox         ;  client parameters
;


MEMSPLIT=64                     ; splitting factor of the histogram memory
MEMPOS=0                        ; position of the first (a-) spectrum
;
DELAY=0.                        ; analyser start delay in us
TCHANNEL=64.                    ; channel width in us
;
CHANNELS=abcdefg                ; symbols of recorded detector channels
RECSTART=1                      ; first time channel to be recorded
RECSTOP=3100                    ; last time channel to be recorded
DATADIR=/h1/usr/user/data/      ; data directory
;
PLATFORM=116.80                 ; NSHR platform angle
PERIOD=Dec97c2                  ; working cycle of the IBR-2 reactor
;
INCIDENT=v0491.w05              ; incident file name for VSP client
;
;       Goni part
;
;--------- SKAT definitions ----------------------------------------------
;STEPMOTOR=      phi,    0.0025,     deg,    800., 0, 1, 0, 0
;STEPMOTOR=      z,      12.5e-3,    mm,     200., 1, 0, 0, 1
;
;-------- NSHR definitions -----------------------------------------------
STEPMOTOR=      phi,    0.36,       deg,    275., 0, 1, 0, 0
STEPMOTOR=      z,      5e-3,       mm,     800., 1, 0, 0, 1
STEPMOTOR= omega, 8.333333e-3,      deg,    400., 1, 1, 1, 1
;
;       end of file
;
```

Литература

1. Зем Ен Кен и др.: Система накопления, управления и контроля спектрометра НСВР в стандарте VME. Сообщение ОИЯИ, Р13-94-73, Дубна, 1994.
2. N.N.Isakov et al.:SKAT-1 a new texture spectrometer at the IBR-2 reactor. Report on XV International Workshop on the Applications of Neutron Scattering to Solid State Physics, 17-23 March 1997, Zarechny, Russia.
3. J.Heinitz, A.S.Kirilov: A software complex for neutron time-of-flight measurements by means of VME based accumulation, control and supervising system. Comm. of the JINR, D13-95-462, Dubna, 1995.
4. EUROCOM-17-5xx Dual 68040 CPU Board with Graphics. Hardware Manual, Revision 1 A, V-E17.-A995, ELTEC Electronik Gmbh.
5.CompControl International B.V.: X Window System X11/OS9 User's Manual, version 1.1, March 1991.

Рукопись поступила в издательский отдел
15 мая 1997 года.

Кирилов А.С., Хайнитц Й. Р13-97-161

Интерпретация процедуры эксперимента
в программном комплексе систем накопления,
управления и контроля спектрометров НСВР и СКАТ
(задача Join)

Для освобождения пользователя от рутинной работы по управлению спектрометром процедура измерения записывается в виде текстового файла, который получил название программы эксперимента. В данной работе рассматриваются общие идеи, принцип организации связи и протокол взаимодействия задачи-интерпретатора с исполнительными клиентами, а также особенности реализации и интерфейс этой задачи.

Достоинствами подхода являются:
— удобство управления спектрометром;
— легкость создания программы эксперимента и внесения в нее изменений;
— универсальность задачи-интерпретатора по отношению и к клиентам, и к установке (спектрометру);
— возможность естественного развития системы путем добавления новых клиентов.

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

Сообщение Объединенного института ядерных исследований. Дубна, 1997

Перевод авторов

Kirilov A.S., Heinitz J. P13-97-161

Experimental Procedure Execution in the Software Complex
for the Accumulation, Control and Supervising Systems
at the NSHR and SKAT Spectrometers
(the JoinTask)

To eliminate routine work to control the spectrometer, a measurement procedure called an experiment program is created as a text file. This article is devoted to the basic ideas, the principle and the communication protocol of an interpreter task and its managed clients, as well as the implementation notes and user interface.

Our approach is characterized by the following main features:
— comfortable spectrometer control;
— ease of making changes in the experimental procedure;
— versality of the interpretation task both for managed clients and for the spectrometer itself;
— ease of extending the complex to handle new experimental equipment.

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

Communication of the Joint Institute for Nuclear Research. Dubna, 1997