

ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

0-774

P10-85-581

А. И. Островной

МЕТОДИКА ПРОГРАММИРОВАНИЯ
СИСТЕМ АВТОМАТИЗАЦИИ ЭКСПЕРИМЕНТОВ
НА ЯЗЫКЕ ПАСКАЛЬ

Направлено в журнал "Автометрия"

1985

Одной из самых острых проблем в области автоматизации научных исследований является разработка программного обеспечения систем автоматизации экспериментов, созданных на базе мини- или микро-ЭВМ и аппаратуры КАМАК. Актуальность этой проблемы не уменьшается, несмотря на значительное количество работ в этой области. Важность её решения следует из большого количества создаваемых систем автоматизации и трудоёмкости разработки программного обеспечения. По некоторым оценкам /1/ стоимость программного обеспечения превышает 50% стоимости системы автоматизации экспериментов в целом.

Повысить производительность труда программистов можно только на основе использования языков высокого уровня. В данной работе описывается методика программирования систем автоматизации спектрометрических экспериментов на языке ПАСКАЛЬ.

В литературе, посвященной проблеме создания программного обеспечения систем автоматизации, обычно рассматривается вопрос о способах программирования аппаратуры КАМАК, хотя это не всегда самая трудная и большая по объему работы задача. Описываемая в данной работе методика предполагает использование набора типовых решений для задач, стоящих перед программистами. Выработка таких решений возможна, т.к. всякая система автоматизации экспериментов должна обеспечить выполнение одного и того же набора функций:

- 1) регистрацию и накопление экспериментальных данных;
- 2) визуализацию и предварительную обработку поступившей информации;
- 3) взаимодействие с оператором и управление системой;
- 4) архивизацию данных.

В качестве таких типовых решений предлагается использовать унифицированную организацию программного обеспечения систем автоматизации экспериментов (программных систем). Представлять алгоритмы обработки в виде параллельных процессов. Программирование операций с аппаратурой КАМАК выполнять с помощью набора процедур в соответствии с предложениями комитета ESCONE /2/. Для архивизации данных и ин-

теграции программы накопления с программами обработки (обеспечения их концептуального единства) использовать простую базу данных ^{/3/}. Помимо этого принять типовую форму диалога "человек - ЭВМ".

Перечисленные типовые решения (составляющие методики) коротко описаны ниже. Они определяют лишь форму решения стоящих перед разработчиками программного обеспечения систем автоматизации задач. Это позволяет устраниваться от их индивидуального для каждой системы решения и сконцентрировать внимание и усилия на прикладных аспектах перечисленных задач. В целом, это позволяет ускорить процесс разработки программного обеспечения систем автоматизации и повысить его качество.

Унифицированная организация программных систем (см. рис.1) включает процедуры обработки прерываний, алгоритмы обработки, представления в виде параллельных процессов, таблицу переменных состояния процессов (ПСР), интерпретатор интерактивных приказов и монитор. Все программные компоненты имеют доступ к глобальным переменным, к базе данных и специальному файлу, в котором хранится информация о текущем состоянии аппаратуры, программной системы, проводимых изменений.

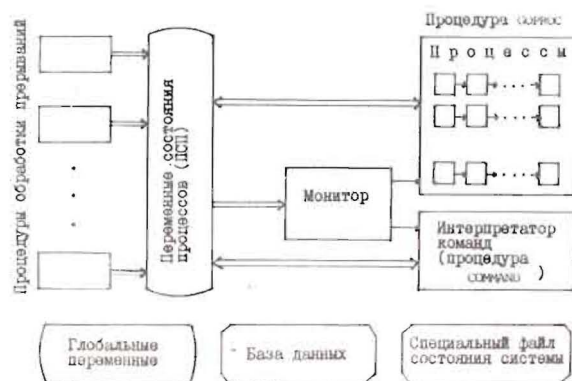


Рис.1. Типовая структура программной системы.

Процедуры обработки прерываний инициируются по прерываниям и обеспечивают передачу информации между оперативной памятью (набором глобальных переменных) и аппаратурой КАМАК. Они выполняют минимально

необходимые действия, а если требуется дальнейшая обработка поступившей информации, то устанавливается флаг готовности в переменной, отражающей состояние нужного процесса. Монитор, просматривая таблицу ПСР, обнаружит готовый к работе процесс и инициирует его. Одновременно монитор обеспечивает прием приказов с терминала пользователя и, если приказ введен, то инициирует интерпретатор.

Параллельные процессы удобно использовать для представления алгоритмов обработки нескольких одновременно функционирующих потоков информации. Это удобно даже, если программа будет исполняться на однопроцессорной ЭВМ. В частности, средства программирования параллельных процессов в таких современных языках, как АЛГА ^{/4/} и МОДУЛА-2 ^{/5/}, также рассчитаны на однопроцессорную вычислительную установку.

В рамках данной методики предлагается простейший способ программирования параллельных процессов, не требующий использования часов ЭВМ или операционной системы с мультипрограммным режимом работы. Они реализуются за счет специальной техники программирования в рамках одной программы.

С точки зрения организации все процессы выполняются одной процедурой GOPROC (см. рис.2). Каждый процесс имеет свой номер, переменную в таблице ПСР и фрагмент процедуры GOPROC. Фрагмент делится на этапы, которые представляют собой неделимые в смысле исполнения "кусочки" программы на языке ПАСКАЛЬ. Переменная, отражающая состояние процесса, содержит информацию о готовности процесса к работе, наличии или отсутствии флага блокировки, ожидании завершения другого процесса, номер очередного этапа, счетчик для вычисления сторожевого временного интервала.

```

PROCEDURE GOPROC (NUP:INTEGER);
... (* ОПРЕДЕЛЕНИЕ ЛОКАЛЬНЫХ ПЕРЕМЕННЫХ *)
BEGIN
IF TIMEOUTFLAG THEN ... (* ОБРАБОТКА TIMEOUT *)
CASE NUP OF
1: BEGIN (* ПРОЦЕСС 1 *)
CASE STAGE OF
1: BEGIN ... END; (* ЭТАП 1 *)
2: BEGIN ... END; (* ЭТАП 2 *)
.
.
.
END; (* КОНЕЦ 1-ГО ПРОЦЕССА *)
2: BEGIN (* ПРОЦЕСС 2 *)
.
.
.
END;
.
.
.
END; (* ОСТАЛЬНЫЕ ПРОЦЕССЫ *)
.
.
.
END; (* CASE NUP... *)
END; (* GOPROC *)

```

Рис.2. Организация параллельных процессов в процедуре GOPROC.

Функционирование процессов осуществляется следующим способом. Монитор в бесконечном цикле проверяет таблицу ПСП, если какой-либо процесс готов к работе, не заблокирован и не ожидает завершения процесса, то монитор обращается к процедуре `GOPROC` с параметром, равным номеру выбранного процесса. `GOPROC` выполняет очередной этап указанного процесса и возвращает управление в монитор, где продолжается ожидание готового к работе процесса.

В сегментах программы, выполняющих отдельные этапы, необходимо предусмотреть изменение состояния процесса (в соответствии с алгоритмом обработки). Например, если процесс завершён, то нужно убрать флаг готовности процесса к работе; если требуется перейти к следующему этапу, то необходимо изменить номер очередного этапа; если же в результате работы данного этапа инициирован другой процесс, то записать номер этого процесса в переменную состояния исходного процесса. Следует отметить, что информацию о состоянии процессов можно задать или изменить в любой части программной системы.

Использование сторожевых временных интервалов обеспечивает монитор. Длительность интервала задается в количестве просмотров таблицы ПСП монитором. По истечении временного интервала монитор устанавливает специальный флаг `TIMEOUTFLAG` и инициирует процедуру `GOPROC` с параметром, равным номеру процесса, для которого истек сторожевой интервал. Таким образом, обеспечивается возможность использования сторожевых временных интервалов с индивидуальной для каждого процесса длительностью и раздельной обработкой.

Такая организация параллельных процессов позволяет реализовать простые, но удобные средства отладки программных систем. Для этого используются интерактивные приказы, позволяющие блокировать исполнение процессов, разблокировать, вывести на терминал информацию о состоянии всех процессов, выполнить алгоритмы, представленные в виде параллельных процессов по шагам. Каждый шаг соответствует выполнению одного этапа выбранного в соответствии с описанным выше алгоритмом процесса. Прерывания на время блокировки не запрещаются и обрабатываются, как обычно. Блокировка не нарушает нормальный ход вычислительного процесса в программной системе, но позволяет его контролировать с помощью интерактивных приказов. Помимо этого, для отладки программных систем могут создаваться специальные приказы, использоваться вывод на терминал или на печатающее устройство промежуточных результатов, а также применяться интерактивный отладчик программ на языке ПАСКАЛЬ.

Программирование операций с аппаратурой КАМАК обеспечивается с помощью набора процедур, аналогичных набору подпрограмм на ФОРТРАНЕ^{6/}

Для реализации этих процедур используются структуры данных, отражающие организацию стандарта КАМАК и тип контроллера, как это предложено в работе ^{7/}. Разработанные структуры данных включают переменные типа вектор, отражающие понятие вектора прерываний в машинах серии СМ ЭВМ, типа регистр для работы с регистрами аппаратуры КАМАК и типа канал, предназначенных для организации обработки прерываний. Переменные типа канал включают адрес процедуры обработки прерываний, индивидуальную маску запросов и адрес вектора прерываний.

Набор процедур для работы с аппаратурой КАМАК содержит процедуры для вычисления значений переменных, описывающих регистры и каналы, процедуры для обеспечения связи вектора прерываний с заготовленной процедурой, выполнения одиночных и блочных операций обмена с блоком КАМАК, разрешения и запрещения работы отдельного канала и всего крейта.

Созданные средства обеспечивают бесконфликтную работу параллельно функционирующих программ обработки прерываний с той же самой аппаратурой КАМАК путём изменения приоритета процессора ЭВМ на время исполнения одной операции КАМАК. Процедура обработки прерываний является обычной процедурой без параметров и может быть реализована на языке ПАСКАЛЬ или ассемблере. Она обеспечивает минимально необходимую обработку и в случае потребности устанавливает флаг готовности в таблице ПСП для последующего инициирования заранее заготовленного процесса обработки.

Использование набора процедур, реализующих работу с аппаратурой КАМАК, позволяет "скрыть" в них детали работы со структурами данных и регистрами контроллера КАМАК. Это облегчает применение предлагаемых средств. Для создания прикладных программ достаточно знать определение упомянутых типов данных, назначение и параметры используемых процедур. Разработанные программные средства обеспечивают использование контроллеров крейта типа IO6 ^{8/} и СС-II ^{9/} на СМ ЭВМ.

Использование простой базы данных в программах накопления и обработки обеспечивает унифицированное представление экспериментальной информации и единообразный доступ к ней со стороны большого числа программ, работающих как в on-line, так и в off-line режимах. Во время проведения измерений база данных используется для накопления разнородных параметров, характеризующих файлы с основной спектральной информацией. По окончании измерений она позволяет проводить автоматическую обработку накопленных данных, извлекая для каждого файла дополнительные параметры, необходимые при обработке.

База данных может также использоваться для хранения истории работы системы автоматизации, информации о действиях оператора и сбоях

оборудования, а в дальнейшем – для анализа качества программно-аппаратной системы, поиска причин случайно возникающих сбоев и т.п.

Для организации базы данных и работы с ней используется пакет процедур ^{/3/}. Проблемная ориентация пакета, его компактность позволяют использовать его в системах автоматизации экспериментов на мини- и микро-ЭВМ. Одно из применений этой базы данных описано в работе ^{/10/}.

Форма диалога "человек-ЭВМ" имеет важное значение ^{/11/}. Особенностью систем автоматизации экспериментов является то, что исполнение приказа часто влечет за собой необратимые действия, связанные с реально происходящими физическими процессами. Вследствие этого программная система должна обеспечить контроль не только синтаксиса приказов, но и допустимость значений вводимых параметров и возможность исполнения приказа в данном состоянии системы. Целесообразно в каждой программной системе реализовать команду HELP (ПОМОГИ), по которой пользователю выводится на терминал список допустимых приказов с пояснениями об их назначении.

Реализовать все приказы предлагается в одной процедуре COMMAND (см. рис.1). Пользователь может вводить имя приказа или в одной строке с именем через запятую перечислить значения параметров. Монитор принимает имя приказа, если далее следуют параметры, то он выработает специальный флаг и после этого инициализирует процедуру COMMAND. Прием параметров и исполнение приказа осуществляется соответствующим фрагментом процедуры COMMAND. Если параметры не были перечислены в одной строке с именем приказа, то в COMMAND следует предусмотреть вывод приглашений для последующего ввода параметров.

Специальный файл, содержащий информацию о состоянии системы, автоматически создается в момент первого запуска программной системы. В дальнейшем работа программной системы начинается с копирования содержимого этого файла в оперативную память, так определяются значения переменной состояния системы (ПСС). Она всегда содержит действительные на текущий момент данные о состоянии системы. При изменении состояния программа изменяет значение нужного поля ПСС и обновляет содержимое специального файла на диске.

Конкретный состав информации специального файла и ПСС зависит от задачи. Например, они могут содержать дату и время начала и конца измерений, номер текущего эксперимента и измерения, состояние измерительной системы (идет накопление или нет), имена файлов для спектрометрической информации и базы данных, информацию о состоянии используемых устройств (положение осей гониометра или передвигающейся планки с образцами) и др.

Использование такого специального файла служит обеспечению живучести системы автоматизации. Он дает возможность организовать продолжение измерений с какого-то "разумного" (имеющего физический смысл) момента, даже после сбоя ЭВМ или временного отключения питания. Конечно, все необходимые данные для такой операции должны храниться в файле на диске.

В заключение следует отметить, что описанная методика использована для создания ряда программных систем на ЭВМ СМ-3, СМ-4 и МЕНА-60 с операционной системой РАФОС. Отдельные положения описанной методики были опробованы ранее в созданной в нашей лаборатории системе САННО ^{/12,13/}. В целом получен положительный опыт применения этой методики.

Использование языка высокого уровня ПАСКАЛЬ для программирования систем автоматизации экспериментов обеспечивает простоту модификации и расширения программных систем, является основой для создания качественного и мобильного программного обеспечения.

Литература

1. Л.А. Маталин-Слудский, И.Ф. Колпаков. "Автоматизированные системы научных исследований и аппаратуры КАМАК" – Природа, 1984, № 2.
2. "Subroutines for CAMAC", ESONE/SR/01 Sep. 1978.
3. А.И. Островной ОИЯИ, РГО-84-439, Дубна, 1984.
4. Я.Пайл "Ада – язык встроенных систем" – Пер. с англ. М., Финансы и статистика, 1984, с.238.
5. Wirth N. Modula - 2. Institut for Informatik, ETH, N.36, Zurich, 1980.
6. О.В. Выхшкин, П.Л. Храпкин. Пакет стандартных подпрограмм для работы с аппаратурой КАМАК. Автометрия, 1982, № 4.
7. Г.Балука, И.М. Саламатин. ОИЯИ, ГО-84-573, Дубна, 1984.
8. Интерфейс СМ-3 – КАМАК типа IO6A, IO6B. Инструкция по обслуживанию 32 776-00000-002. ZZUJ "POLON", ZPE, 02-673, Warszawa.
9. Ball F. et al. CERN-NP, CAMAC Note No - 43-00, Geneva, 1973
10. А.М. Балагуров и др. ОИЯИ, РГО-84-440, Дубна, 1984.
11. В.Денниг; Г.Эссинг, С.Масс. Диалоговые системы "человек – ЭВМ" Адаптация к требованиям пользователя. Пер. с англ. М., Мир, 1984, с.112.
12. А.И. Островной. Монитор для специализированных систем автоматизации экспериментов. В кн.: Структура, технические средства и организация систем автоматизации научных исследований. (Материалы XV Всесоюзной школы по автоматизации научных исследований). ДИЯФ, Л., 1982, с.106-116.
13. А.И. Островной, И.М. Саламатин. ОИЯИ, РГО-81-342, Дубна, 1981.

Рукопись поступила в издательский отдел
30 июля 1985 года

Принимается подписка на препринты и сообщения Объединенного института ядерных исследований.

Установлена следующая стоимость подписки на 12 месяцев на издания ОИЯИ, включая пересылку, по отдельным тематическим категориям:

ИНДЕКС	ТЕМАТИКА	Цена подписки на год
1.	Экспериментальная физика высоких энергий	10 р. 80 коп.
2.	Теоретическая физика высоких энергий	17 р. 80 коп.
3.	Экспериментальная нейтронная физика	4 р. 80 коп.
4.	Теоретическая физика низких энергий	8 р. 80 коп.
5.	Математика	4 р. 80 коп.
6.	Ядерная спектроскопия и радиохимия	4 р. 80 коп.
7.	Физика тяжелых ионов	2 р. 85 коп.
8.	Криогеника	2 р. 85 коп.
9.	Ускорители	7 р. 80 коп.
10.	Автоматизация обработки экспериментальных данных	7 р. 80 коп.
11.	Вычислительная математика и техника	6 р. 80 коп.
12.	Химия	1 р. 70 коп.
13.	Техника физического эксперимента	8 р. 80 коп.
14.	Исследования твердых тел и жидкостей ядерными методами	1 р. 70 коп.
15.	Экспериментальная физика ядерных реакций при низких энергиях	1 р. 50 коп.
16.	Дозиметрия и физика защиты	1 р. 90 коп.
17.	Теория конденсированного состояния	6 р. 80 коп.
18.	Использование результатов и методов фундаментальных физических исследований в смежных областях науки и техники	2 р. 35 коп.
19.	Биофизика	1 р. 20 коп.

Подписка может быть оформлена с любого месяца текущего года.

По всем вопросам оформления подписки следует обращаться в издательский отдел ОИЯИ по адресу: 101000 Москва, Главпочтамт, п/я 79.

Островной А.И.

P10-85-581

Методика программирования систем автоматизации экспериментов на языке Паскаль

В работе описана методика программирования систем автоматизации экспериментов на языке Паскаль. Она основана на использовании типовых решений для задач, стоящих перед программистами. В их число входят унифицированная организация программных систем, представление алгоритмов обработки в виде параллельных процессов, использование набора процедур для программирования операций аппаратуры КАМАК, применение простой базы данных для архивизации экспериментальной информации. Методика использовалась при создании ряда программных систем на ЭВМ СМ-3, СМ-4, МЕРА-60 с операционной системой РАФОС.

Работа выполнена в Лаборатории нейтронной физики ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1985

Перевод О.С.Виноградовой

Ostrovnoj A. I.

P10-85-581

Method for Experiment Automation Programming in Pascal Language

A method for experiment automation system programming in high-level Pascal language is described. It is based on using unified solutions of problems for programmers to solve. They include unified structure of programs, data processing algorithms as parallel processes, using of sub-routine package for CAMAC programming and simple data base for archiving experimental information. The method was used for development of software for some experiment automation and spectra unfolding systems based on PDP-11 like mini- and microcomputers with RT-11 operational system.

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

Preprint of the Joint Institute for Nuclear Research. Dubna 1985