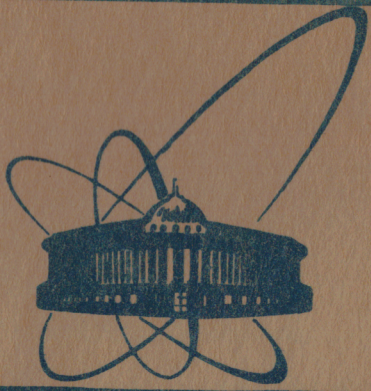

сообщения
объединенного
института
ядерных
исследований
дубна

Р10-80-424

Г.Балука, Г.П.Жуков, Ю.Намсрай, А.И.Островной,
А.С.Савватеев, И.М.Саламатин

МЕТОД НАСТРОЙКИ В ПРИМЕНЕНИИ К ПРОГРАММИРОВАНИЮ РАБОТЫ ЭВМ НА ЛИНИИ С ЭКСПЕРИМЕНТАЛЬНЫМ ОБОРУДОВАНИЕМ

1. Формулировка подхода

1980

Балука Г. и др.

Р10-80-424

Метод настройки в применении к программированию работы ЭВМ на линии с экспериментальным оборудованием. 1. Формулировка подхода

Описан метод генерации прикладных программ для работы с экспериментальным оборудованием /ЭО/ из унифицированных программных модулей /УПМ/, настраиваемых на конкретную конфигурацию оборудования перед использованием в эксперименте.

Данный подход по сравнению с двухуровневым подходом обеспечивает увеличение быстродействия программ за счет устранения избыточного уровня косвенной адресации.

Для реализации метода разработан язык высокого уровня для описания работы с ЭО, компилятор, компонующие программы и библиотеки УПМ, используемых в перемещаемом формате загрузки.

Программы реализованы на машинах с системой команд СМ-3 для программирования экспериментов на реакторе ИБР-2.

Работа выполнена в Лаборатории нейтронной физики ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1980

Baluka G. et al.

P10-80-424

A Method of Setting Applied to Programming Computer On-Line with Hardware. 1. Formulation

I. ВВЕДЕНИЕ

Развитие технологии производства электронного оборудования обусловило интенсификацию использования его для автоматизации научных исследований. Снижение стоимости мини-, а особенно микро-ЭВМ привело к тому, что такие ЭВМ становятся наиболее распространенными устройствами управления при проведении ядерно-физических и других экспериментов. Соединенное с ЭВМ экспериментальное оборудование (ЭО) может быть эффективно использовано только после создания соответствующего прикладного программного обеспечения. Быстрый рост потребности в таких прикладных программах выдвигает задачу развития средств и методов программирования экспериментальных установок в число актуальных. Программное обеспечение эксперимента обычно включает программу (или систему), обслуживающую экспериментальное оборудование, программы обработки данных и другие.

Программа, работающая с ЭО, является сложной и ответственной компонентой прикладного программного обеспечения. В число решаемых ею задач входят диспетчеризация и обработка прерываний, ввод и вывод информации - иначе говоря, регистрация, ввод и накопление экспериментальных данных, контроль состояния и управление экспериментальной установкой.

Разработка стандартных средств подключения экспериментальной аппаратуры к ЭВМ (например, стандартов КАМАК /1,2/, IEEE-488 /3/ и других) существенно облегчила задачу программирования работы ЭВМ на линии с ЭО. Тем не менее в настоящее время стоимость программ остается значительной. В то же время наблюдается диспропорция в соотношении числа разработок, посвященных ЭО, и его программному обеспечению. Так, из 450 наименований, приведенных в работе /4/, только 50 относятся к программным средствам.

Авторами были разработаны средства программирования взаимодействия ЭВМ с экспериментальным оборудованием в реальном масштабе времени. В их состав входят язык, служебные программы для генерации прикладной программы и пакет унифицированных программных модулей. Согласно введенной методике по мере появления в оснащении лаборатории новых блоков оборудования создаются унифицированные программные модули для выполнения типичных прикладных операций с этим оборудованием. В числе таких модулей имеются, например, программы гистограммирования ("анализатор"), регистрации событий (одно- и многопараметровых), регистрации с двойной буферизацией данных, управления устройствами перемещения образцов, ориентации их в пространстве, драйверы дисплеев и другие. Эти модули, предварительно отлаженные, заносятся в библиотеки в виде, не зависящем ни от способа подключения оборудования (например, от номера станции в каркасе), ни от методики конкретного эксперимента. Модули с такими свойствами могут быть использованы без изменения в разных экспериментах, в которых требуются аналогичные прикладные операции. При необходимости создания программного обеспечения для конкретного эксперимента на специализированном языке описываются состав оборудования, способ его подключения к ЭВМ, состав и последовательность выполнения прикладных операций. Служебные программы на основании этого описания собирают из библиотек запасенные модули, объединяют их, настраивают и, в конечном счете, создают программное обеспечение эксперимента с заданными функциями. В данном цикле работ описываются разработанные средства.

Цель разработки - сократить сроки создания и стоимость программного обеспечения экспериментов на реакторе ИБР-2 /5/. Реализация выполнена для мини- и микро-ЭВМ типа СМ-3 /6/, "Электроника-60" /7/ в рамках проекта, описанного в работе /8/.

Данное сообщение посвящено формулировке и обоснованию выбора подхода к решению задачи.

2. КРАТКИЙ ОБЗОР

Задача программирования работы с экспериментальным оборудованием, помимо описания алгоритма работы, включает описание конфигурации оборудования.

Подход к решению этой задачи может быть охарактеризован:
I) используемыми языковыми средствами;
2) способом обработки параметров, описывающих конфигурацию оборудования и адресованные ему команды;
3) способом компоновки прикладной программы из модулей или подпрограмм.

Языковые средства и способ компоновки прикладных программ влияют на круг пользователей (уровень необходимой квалификации в области программирования) и технологичность методики программирования. Способ обработки параметров определяет временные характеристики программ (их вклад в мертвое время и время реакции).

2.I. Известные разработки , посвященные программированию работы с ЭО, выполнены в рамках следующих основных подходов:

I) создание подпрограмм для реализации прикладных операций с конкретной конфигурацией оборудования (т.н. прагматический подход /9/);

2) реализация команд используемого стандарта оборудования в виде набора подпрограмм (например, /I0/);

3) расширение существующего языка операторами, ориентированными на команды используемого стандарта ЭО /II-I3/;

4) создание нового языка программирования /I4,I5/.

В прагматическом подходе программируются непосредственно те операции, которые нужны для решения конкретной задачи. Такой подход обладает рядом достоинств /9/, которые обеспечили ему широкое распространение. Реализация его, как правило, возможна без дополнительных системных разработок. Однако такой подход характеризуется минимальным обеспечением преемственности программного обеспечения и гибкости его по отношению к конфигурации оборудования. Тем самым экономичность этого метода в условиях разработки небольшого числа простых программ обращается в свою противоположность по мере возрастания суммарного объема программ.

Остальные подходы перечислены в порядке возрастания дополнительных затрат, необходимых для реализации методики программирования. Эти подходы подробно описаны, например, в обзорах /I6,I7/.

2.2. Для указанных направлений разработчики средств программирования и прикладных программ использовали указанные ниже способы работы с параметрами, описывающими конфигурацию оборудования:

I) компиляцию параметров (адресов регистров, векторов прерывания и др.), при которой абсолютные адреса регистров оборудования вставляются в текст программы в формате загрузки;

2) интерпретацию параметров во время исполнения программы, например, вычисление адресов оборудования непосредственно перед выполнением адресованных ему команд;

3) так называемый "двухуровневый подход" /I3,I8/, при котором интерпретация параметров выполняется один раз перед исполнением прикладной программы, после чего остается один уровень косвенной адресации, избыточный по сравнению со способом компиляции параметров.

Очевидно, что первый способ позволяет получить программы с наилучшими временными характеристиками, второй – наиболее медленные программы, а третий способ является промежуточным в этом отношении.

2.3. Компоновка прикладных программ осуществлялась средствами редактора связей или с помощью специализированных генераторов /I9,20/.

Редактор связей является наиболее распространенным и универсальным средством компоновки. Он позволяет связать несколько программных модулей в одну прикладную программу, используя глобальные имена, описанные в различных модулях. Недостатком этого способа следует считать необходимость определять взаимосвязи между модулями на этапе их проектирования и программирования.

Как следствие этого, средства компоновки с помощью редактора связей не обеспечивают преемственности развития прикладного программного обеспечения, затрудняют модификацию и сопровождение отдельных программных модулей, не обеспечивают возможности оперативной перекомпоновки модулей внутри прикладной программы. Для использования таких средств компоновки необходимо знать структуру, форматы данных и способ взаимодействия модулей, поэтому пользователями редакторов связей выступают, как правило, авторы тех программных модулей, из которых составляется прикладная программа. При создании программ для работы с ЭО пользователями этих средств компоновки являются программисты.

Способ генерации прикладных программ позволяет повысить уровень средств компоновки и благодаря этому существенно облегчить использование их потребителями прикладных программ, т.е. специалистами, непосредственно заинтересованными в результатах компонов-

ки. Генерация прикладных программ выполняется, как правило, для какой-то узкой проблемной области и может иметь различные модификации /21,22,23,24/. Общим для различных методов генерации является использование какого-то стандарта при программировании модулей и описании взаимосвязей между ними /21,8/. Метод генерации позволяет модифицировать отдельные программные модули без изменения остальной части прикладной программы; легко и быстро модифицировать способ соединения модулей и состав прикладной программы; предоставить средства компоновки пользователям прикладных программ; способствует эволюционному развитию прикладного программного обеспечения.

3. УСЛОВИЯ РАЗРАБОТКИ И ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

3.1. Выбор направления развития средств программирования определяется особенностями подхода, а также целями и условиями разработки и эксплуатации этих средств.

Приведенную во введении формулировку цели можно дополнить ссылкой на ожидаемое число прикладных программных систем (около 30) и нашу оценку необходимых сроков их создания (3 –5 лет).

Наши условия являются обычными для исследовательской организации. Имеется в виду

1) довольно большое число необходимых различных экспериментальных установок;

2) малый "срок жизни" прикладных программ (обычно в пределах 1 или нескольких лет);

3) настоятельная потребность уменьшить до сроков порядка месяца и менее разрыв во времени между моментом формулировки функций прикладной программы и вводом ее в эксплуатацию (включая и время обучения пользователя);

4) неоднократные модификации программ (и конфигурации оборудования) за время их жизни;

5) уникальность каждой экспериментальной системы;

6) сравнительно небольшое число (десятки) типовых модулей оборудования, из которых компонуются экспериментальные системы;

7) случайность конфигурации ЭВМ, используемой в составе экспериментальной установки, в отношении объема ОЗУ, по крайней мере – неоптимальность;

8) обычно малый (~ 10) численный состав коллектива программистов.

3.2. Эти условия определяют требования

1) обеспечения скоростных характеристик (мертвое время и время реакции /25/) программ, соответствующих наиболее жестким требованиям в данной проблемной области либо по крайней мере не ухудшающих характеристик используемого оборудования. Заметим, что у большинства мини-ЭВМ минимальное возможное время холостого цикла в сумме с временем диспетчеризации запросов прерывания, достигающее 30÷100 мкс, уже превышает характерные времена доступного экспериментального оборудования, составляющие 1÷50 мкс;

2) гибкости программного обеспечения по отношению к составу и конфигурации ЭО;

3) преемственности разработок, в первую очередь – преемственности проверки, отладки, а также облегчения сопровождения прикладных программ;

4) организации ввода и вывода данных и управляющих сигналов одновременно по нескольким независимым каналам;

5) принятия специальных мер по обеспечению документацией, необходимой для разработки и эксплуатации программ.

Разработка силами прикладных программистов специальных средств, совершенствующих технологию программирования, может быть оправдана, если суммарный расход сил и времени на создание таких средств и самих прикладных программ не превысит расхода, ожидаемого в случае использования только средств операционных систем общего назначения. Экономическую целесообразность разработки заранее количественно определить затруднительно. Мы полагаем, что в данных условиях целесообразно максимально использовать доступные средства систем программирования общего назначения. Однако в нужной мере удовлетворить перечисленным требованиям средствами таких ОС (например, RT-II /26/) не представляется возможным. По этой причине была выполнена описываемая разработка.

4. ФОРМУЛИРОВКА ПОДХОДА

Описываемый в данном сообщении подход опирается на разработанную структуру унифицированных программных модулей (УПМ), способ представления их в перемещаемом формате REL /27/, способ настройки модулей в соответствии с конкретной конфигурацией (адресами) оборудования и метод генерации прикладных программ из унифицированных модулей.

4.1. В виде унифицированных модулей программировались типичные для обслуживаемой проблемной области прикладные операции в

условных адресах регистров оборудования. В этом плане данный подход обладает всеми достоинствами, присущими прагматическому подходу.

Унификация программных модулей достигнута путем разработки единой структуры для модулей, обслуживающих оборудование, и интерфейса для обмена данными и управлением с организующими программами и программами обработки данных.

Программный модуль содержал следующие элементы структуры:

1) группу входов для выполнения основного алгоритма, вспомогательных операций и управления потоком информации. Назначение входов, их порядок и способ доступа к ним были фиксированы;

2) описание элементов данных;

3) функциональные части, отвечающие отдельным входам;

4) таблицу адресов кодов в функциональных частях для настройки команд в соответствии с конкретной конфигурацией оборудования.

Интерфейс для обмена данными и управлением между УПМ и организующими программами был построен следующим образом. Управление УПМ получали двумя способами в зависимости от их типа. УПМ, предназначенные для обработки прерываний от оборудования, подучали управление от диспетчера прерываний и заканчивали работу командой выхода из прерывания. Эти модули использовались для компоновки резидентной части подсистемы, работающей с ЭО. УПМ, работающие с оборудованием без прерывания, оформлялись в виде подпрограмм типа SUBROUTINE и представляли собой стандартные программы /8/.

Для всех УПМ сформулирован стандартный способ доступа непосредственно к элементам данных (ЭД) в оперативной базе данных /8/. Соответствующие ЭД содержали исходные данные либо результаты работы УПМ.

4.2. Настройка программного модуля выполнялась в два этапа. На одном этапе настройка обеспечивала работоспособность перемещаемой программы на занятом ею участке оперативной памяти. Такая настройка подробно описана в работах /27,28/. На другом – выполнялась подмена условного кода в адресной части машинной команды абсолютным адресом с целью обеспечения работоспособности программы с "перемещаемым" оборудованием для конкретного варианта его подключения. Алгоритмы составлены так, что после настройки обе таблицы перемещающей информации для работы программ не требовались и могли не присутствовать в оперативной памяти. Мы сохраняли в теле настроенного модуля таблицу, использовавшуюся для настройки его на конкретные адреса оборудования, и тем самым была обеспечена возможность изменить настройку модуля в соответствии с приказом оператора в интерактивном режиме.

Особенностью данного метода работы с адресами оборудования, отличающей его от описанного в работе /I8/, является отсутствие одного избыточного уровня косвенной адресации, что привело к улучшению мертвого времени программ.

4.3. Реализация подхода существенно упростилась благодаря унификации интерфейса и структуры программных модулей.

Для создания унифицированных программных модулей использовались готовые компиляторы MACRO-II и FORTRAN-IУ, редактор связей, редактор текстов, программа отладки и другие программы в рамках операционной системы общего назначения RT-II /26/. Столь существенная ориентация на средства ОС общего назначения значительно сократила объем дополнительных методических разработок.

Разработанные средства генерации включали язык генерации, компилятор, компонующую программу, программы управления библиотеками /3I/ и библиотеки УПМ.

На рисунке показана схема процедуры генерации. Тексты описания на языке САНПО /8/ прикладных программ, работающих с экспериментальным оборудованием, включали:

I. Описание оборудования. Это описание содержало название стандарта, таблицу соответствия адресов и векторов прерывания, символические имена блоков с указанием занятых ими адресов. Изменяемые части адресов регистров блоков указывались в понятиях, свойственных стандарту оборудования (например, B, C и N для стандарта КАМАК /I,2/; адреса памяти для стандарта DEC /29/ и т.д.)

2. Описание способа взаимодействия программ и оборудования, включая наименования обслуживающих программных модулей и списки их параметров (элементов данных).

На основании такого описания компилятор вырабатывал списки программных модулей и другую служебную информацию, необходимую для работы компонующих программ.

Компонующие программы с помощью средств управления библиотеками собирали из библиотек нужные модули, настраивали их для работы с заданной конфигурацией оборудования и формировали прикладные программы в виде загрузочных модулей в перемещаемом формате REL /27/. Помимо этого, по соответствующей инструкции могла быть сформирована прикладная программа в абсолютном формате LDA /27/. Все модули, которые требовались для формирования прикладной программы, заблаговременно создавались и заносились в библиотеки /30/. В дальнейшем эти модули использовались без изменения в прикладных программах для различных экспериментов.

ЗАКЛЮЧЕНИЕ

К достоинствам данного подхода можно отнести следующие:

1) для программирования унифицированных программных модулей используются готовые компиляторы;
2) достигается предельное быстродействие программ путем создания компилируемых программных модулей ;
3) обеспечивается преемственность результатов отладки УПМ;
4) реализуется прагматический подход, обеспечивающий удобное разделение труда, при котором программисты создают библиотеки модулей, пользователи описывают область приложения;
5) облегчается создание машинно-независимых программ /I8/;
6) использование унифицированных модулей становится доступным для специалистов, не искушенных в программировании.

Отличительными особенностями подхода являются:

I. Метод настройки программ в соответствии с конкретными адресами оборудования. Принятое решение устранило в командах, обращенных к оборудованию, избыточный уровень косвенной адресации, характерный для двухуровневого подхода /I8/, и тем самым обеспечило сокращение мертвого времени программ обработки прерываний;

2. Унифицированные структура модулей и интерфейс, обеспечивающие преемственность программных модулей.

Основной результат работы мы видим в том, что разработанные средства позволили реализовать технологию программирования экспериментального оборудования, обеспечившую преемственность отладки УПМ и сокращение сроков создания программ экспериментов.

В заключение пользуемся случаем поблагодарить коллег за многие полезные обсуждения, В.А.Вагова за помощь в работе с оборудованием, Т.Б.Журавлеву за оформление рукописи.

ЛИТЕРАТУРА

1. CAMAC. A.Modular Instrumentation System for Data Handling. Revised Description and Specification. EUR 4100e, 1972.
2. CAMAC. Organisation of Multi-Crate Systems Specification of the Branch Higway and CAMAC Crate Controller Type A. EUR 4600e, 1972.
3. Loyghry D.C. A new instrumentation interface: needs and progress toward a standard. ISA Trans., v.14, No.3, 1975, p.225-230.
4. Stuckenberg H.I. CAMAC Bibliography. CAMAC Bull. No.13,Sept, 1975.
5. Ананьев В.Д., Блохинцев Д.И., Булкин Ю.М. и др. ИБР-2 - импульсный реактор периодического действия для нейтронных исследований. Препринт ОИЯИ, Р3-10888, Дубна, 1977.
6. Наумов Б.Н., Боярченков М.А., Кабалевский А.Н. Управляющий вычислительный комплекс СМ-3. Приборы и системы управления, № 10, 1977, стр.12-15.
7. Борисенко В.Д., Плотников В.В., Талов И.Л. Микро-ЭВМ "Электроника-60". Электронная промышленность, № 10 (70), 1978, стр. 20-21.
8. Балука Г., Жуков Г.П., Намсрай Ю. и др. Комплекс средств для генерации прикладных систем автоматического накопления и предварительной обработки данных - САНПО. Сообщение ОИЯИ, 10-12960, Дубна, 1979.
9. Michelson I., Halling H. Procedure Calls - a Pragmatic Approach. CAMAC Bull., No. 9 Apr. 1974.
10. Thomas R.F. Specifications for Standard CAMAC Subroutines. CAMAC Bull., No.6 March 1974, p.23-26.
11. Нойберт П. О реализации предварительного компилятора для языка КАМАК в ОС ЭВМ НР-2116С. Сообщение ОИЯИ, 11-10279, Дубна, 1977.
12. CAMAC: Proposal for a CAMAC Language by the ESONE Committee sofware Working Group. CAMAC Bull, No.6, 1972.

13. CAMAC: The Definition of IML, a Language for use in CAMAC Systems. ESONE /IML/ 01,ESONE Secretariat, Oct. 1974, 36p.
14. Kappatch A. PEARL,Surwey of Language Features. Karlsruhe, 1977, /KFK-PDV 141/ 18p.
15. RTL/2 Language Specification. London W1P 1 HF, 1974, 55p.
16. Бредихин С.В., Песляк П.М. Средства программирования для CAMAC. Автометрия № 4, 1974, стр. 39-50.
17. Clout P.N. CAMAC Software. IEEE Trans. on Nucl.Sci.,NS-24,No.1, 1977, p. 484-488.
18. Halling H., Michelson J. Will Fortran Tolerate with IML? SWG 20/72, 1972.
19. Баррон Д. Ассемблеры и загрузчики. М., Мир, 1974.
20. Система математического обеспечения ЕС ЭВМ. Под общ. ред. Ларионова А.М. М., Статистика, 1974.
21. Говорун Н.Н., Дорж Л., Иванов В.Г., Лукьянцев А.Ф. Принципы организации и структура модульной системы программ обработки экспериментальных данных. ЭЧАЯ, М., Атомиздат, т.6, вып.3, 1975, с.743.
22. Жук В.И., Шитиков Б.И., Модульное программирование на разговорном языке описания блок-схем. Автометрия, № 1, 1976.
23. Талыкин Э.А., Модульное программирование в задачах сбора и обработки экспериментальных данных. Автометрия, № 1, 1976.
24. Taff L.M. Sporrel F. Flexible real-time nuclear data acquisition processing sofware. Nucl. Instr. and Meth., v.160, No.1, 1979, p.147-158.
25. Дади К., Дади Л., Жуков Г.П. и др. Структура программного обеспечения измерительного модуля для ИБР-2. Сообщение ОИЯИ, 10-9060, Дубна, 1975.

26. RT-11 System Reference Manual (DEC-11-ORUGA-C-D). DEC, Maynard, Massachusetts, 1975, 985p.

27. RT-11 Software Support Manual (DEC-ORPGA-B-D). DEC Maynard, Massachusetts, 1975, 342p.

28. Балука Г., Островной А.И. Динамическое распределение памяти в системе САНПО для ЭВМ типа СМ-3. Сообщение ОИЯИ, Р10-13004, Дубна, 1979.

29. PDP-11 Peripherals Handbook. DEC, Maynard, Massachusetts,1975, 634p.

30. Балука Г., Саламатин И.М., Хрыкин А.С. Организация библиотеки перемещаемых программ для машин типа М-400. Сообщение ОИЯИ, 10-12545, Дубна, 1979.

Рукопись поступила в издательский отдел
18 июня 1980 года.