

сообщения
объединенного
института
ядерных
исследований
дубна

969/84

13/II-84

10-83-761

Г.Балука

ПРОГРАММА-КОМПОНОВЩИК
СИСТЕМ АВТОМАТИЗАЦИИ ЭКСПЕРИМЕНТА
В КОМПЛЕКСЕ САНПО

1983

ВВЕДЕНИЕ

При современном уровне распространения вычислительной техники быстро растут количество и размеры программ и систем автоматизации эксперимента /САЭ/ в исследовательских организациях. В связи с этим становятся все более актуальными разработки средств и методов, обеспечивающих уменьшение затрат на создание САЭ при одновременном улучшении их эксплуатационных характеристик.

Операционные системы предоставляют удобные средства для объединения отдельных компонентов /например, подпрограмм/ в программы и для организации автоматического исполнения заданной последовательности программ, что позволяет строить громоздкие программы и комплексы, реализующие весьма сложные алгоритмы. Однако если программный продукт вырастает до таких размеров, что наличный ресурс оперативной памяти становится недостаточным, возникают специфические проблемы:

1. Скоростные характеристики создаваемой системы требуют высокой квалификации программиста.
2. Стоимость разработки существенно возрастает.
3. Стоимость модификации может непредсказуемо возрастать и зависит от особенностей структуры исходных программ.

В работе /1/ описана система модульного программирования, позволяющая прогнозировать характеристики САЭ, сроки ее создания и обуславливающая простоту модификации.

Это обеспечено специальным способом компоновки САЭ из компонентов, не зависящих друг от друга.

В настоящей работе обосновывается предлагаемый подход к данной проблеме, а также описываются алгоритмы компоновки САЭ и программа-компоновщик.

2. АНАЛИЗ УСЛОВИЙ И ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ К СРЕДСТВАМ КОМПОНОВКИ САЭ

В этом разделе сформулируем требования к средствам программирования и характеристикам САЭ.

Данная работа возникла в связи с интересом к легко реализуемым средствам, ускоряющим процесс производства и последовательного развития силами группы программистов значительного числа программных систем автоматизации эксперимента, функциональные возможности которых перекрываются /1/.

Необходимо также учитывать особенности структуры САЭ, обусловленные ее размерами. При формулировании критерия для клас-

сификации САЭ очень часто определяющими являются количество и ассортимент электронного оборудования, т.е. разрядность и объем памяти ЭВМ, количество устройств внешней памяти, устройств связи, датчиков и др.^{/2/}. В действительности объем оборудования не однозначно связан с количеством и качеством программ, необходимых для обеспечения его работоспособности. При рассмотрении вопросов программирования целесообразно ввести иной способ классификации САЭ, который позволил бы непосредственно разграничивать системы в зависимости от характеристик программного обеспечения /объема, структуры и т.д./.

Будем считать малыми системы, программное обеспечение которых полностью помещается в оперативную /или постоянную/ память ЭВМ, а внешнее запоминающее устройство /если оно включено в состав САЭ/ используется только для первоначальной загрузки программы перед началом эксперимента и для хранения полученных во время работы программы данных. Большими назовем системы, в которых для работы программы обслуживания эксперимента требуется ресурс внешней памяти для хранения программ, принимающих участие в обеспечении работы САЭ.

Спецификой реальных условий использования САЭ является непрерывный рост требований к ней, ввиду чего САЭ систематически развивается. Практика такова, что почти любая САЭ после нескольких этапов модификации исчерпывает ресурс оперативной памяти, после чего либо резко усложняется развитие САЭ, либо приходится строить большую систему. Заметим, что большие системы при соответствующем подходе к их построению обеспечивают возможность неограниченного роста количества обслуживаемых программ. Неизбежные ограничения возникают только из-за конечной емкости ВЗУ и фиксированной максимальной длины отдельного загружаемого модуля.

Существенной особенностью структуры большой САЭ является наличие ядра и группы исполнительных модулей, работой которых управляет ядро. Назовем открытой систему автоматизации эксперимента, способ построения которой допускает изменение состава исполнительных программ без модификации ядра системы. Чаще строятся закрытые САЭ, т.е. системы, при наращивании которых обязательно редактируется ядро.

В таблице, представленной ниже, показаны особенности способа создания закрытых САЭ с использованием фирменных средств программирования, а также некоторые существенные свойства этих систем. Ряд свойств, присущих операционным системам последних лет и практически всем новым языкам /например, MODULA-2^{/3/}, PASCAL^{/4/}, ADA^{/5/} /, таких, как возможность раздельной трансляции сегментов программ, использование языковнезависимого формата программ и библиотек /например, формата .OBJ в системах ФОБОС^{/6/}, ОСРВ^{/7/} для ЭВМ СМ-3, СМ-4 и др./, существенно облегчает создание больших систем. Однако оптимальные средства программирования, ориентированные на близкие по функциональным

возможностям САЭ, отличаются от средств, предоставляемых фирмами массовому пользователю.

	: Закрытые системы	Открытые системы
Способ построения	Использование фирменных /стандартных/ трансляторов и редактора связей.	Использование специальной программы, компоновшей продукт из заранее заготовленных модулей.
Степень преемственности	Требуется редактирование резидентной части	Редактор не требуется
Способ включения /или замены/ перекрывающихся сегментов САЭ	Редактирование исходного текста САЭ	Включение модуля в библиотеку
Способ выделения памяти для перекрывающихся сегментов	Ограничена возможность работы перекрывающихся сегментов. Автор системы распределяет память на этапе разработки.	Автоматически исполняемое динамическое распределение памяти как на этапе компоновки САЭ, так и на этапе исполнения программы /т.е. в процессе эксперимента/
Верификация работоспособности после модификации.	Требуется отладка и проверка САЭ.	Достаточна автономная проверка модуля.

В этой же таблице для открытых систем указан один из возможных способов их создания и приведены прогнозируемые характеристики САЭ.

Из таблицы видно, что ценой создания специальных средств построения открытых систем может быть достигнута существенная экономия за счет повышения технологичности средств программирования. Основной функцией процедуры компоновки САЭ является объединение в один файл управляющей части прикладной системы /монитора/, совокупности часто используемых /во время работы системы/ исполнительных модулей, различных таблиц данных для управления экспериментов и реализации связей между компонентами. Если принять одинаковым формат и способ организации библиотек для модулей, включаемых в тело прикладной системы во

время компоновки, и для модулей, загрузка которых выполняется во время работы САЭ, то такие библиотеки можно объединить /8.9/. Набор модулей, которые включаются в САЭ во время ее компоновки, назовем конфигурацией, управляемой пользователем /КУП/, поскольку он определяется автором системы. Набор модулей, находящихся в ОЗУ во время работы САЭ, будем называть конфигурацией, управляемой автоматически /КУА/, если этот набор отличается от КУП составом или способом размещения модулей и меняется в зависимости от условий работы. Объединение и унификация библиотеки модулей облегчает модификацию КУП, потребность в которой возникает в процессе развития САЭ. После выборки КУП из библиотеки рабочих модулей и включения их в тело системы остальные модули составляют базу для будущей КУА. Помимо облегчения модификации КУП унификация библиотек дает возможность перераспределять память между полями, занимаемыми КУП и КУА, в момент построения САЭ программой-компоновщиком, а также во время работы САЭ - автоматически. Положительным свойством второго режима работы является возможность все поле рабочих программ отдать большой программе. Такая динамика не освобождает автора системы от заботы о том, чтобы после определения КУП на рабочем поле осталось достаточно места для размещения самой длинной из программ, оставшихся для загрузки в динамическом режиме.

В последующих параграфах рассмотрим алгоритм компоновки САЭ и особенности его реализации исходя из требований, перечисленных в таблице для открытых систем.

3. АЛГОРИТМ КОМПОНОВКИ

Суть процесса компоновки состоит в объединении в систему отдельных компонентов. САЭ в системе САНПО включает управляющую программу-монитор, группу исполнительных программ, предназначенных для выполнения прикладных операций, и описание последовательности прикладных операций /алгоритм эксперимента/, интерпретируемое монитором /1/.

Основные идеи, на которые опирается данная работа, следующие:

- 1/ обеспечить возможность раздельного построения трех перечисленных компонентов САЭ;
- 2/ обеспечить возможность динамически во время работы САЭ /т.е. во время эксперимента/ изменять состав функциональных программ, присутствующих в ОЗУ ЭВМ.

Рассмотрим некоторые детали структуры САЭ и алгоритмы, реализующие такие свойства. Основных структурных особенностей две:

- 1/ функциональные программы, используемые для компоновки САЭ, обработаны редактором связей и имеют формат загрузки /в системе ФОБОС/6/ формат ,REL/;
- 2/ после того, как определен алгоритм эксперимента, в предоставленной для САЭ оперативной памяти может быть выделен опре-

деленный участок, предназначенный для размещения функциональных программ. Назовем его динамически распределяемой памятью.

На рис.1 показана упрощенная схема алгоритма компоновки. Положение процесса компоновки в общем процессе создания САЭ показано на рис.2. Программе-компоновщику передается перечень функциональных программ, которые должны быть включены в САЭ, и название содержащей их библиотеки. Компоновщик вычисляет ряд необходимых параметров, по списку выбирает программы из библиотеки, размещает их друг за другом, начиная с одного края поля ДРП, а с другого края располагает таблицу характеристик /ТХ/ этих программ, включающую имена и адреса входов в программу. При такой конструкции поля ДРП программа-интерпретатор получает доступ к любой из программ на ДРП, если ей известен адрес начала ТХ.

Заказ на компоновку возникает в результате разбора соответствующими программами описания САЭ, составленного пользователем и содержащего прямое его указание на то, какие программы поместить на ДРП. В результате размещения указанных программ на ДРП возникает конфигурация, управляемая пользователем. Заметим, что в описанном алгоритме пользователь лишь управляет этой конфигурацией, меняя перечень программ/в отличие, например, от случая использования оверлейных структур, когда автор системы вынужден еще сам распределить память между сегментами/.

Когда длина ДРП недостаточна для размещения всех принадлежащих САЭ функциональных программ, САЭ становится большой. Предположим, что в любой большой САЭ автор может выделить функциональные программы, для которых величина времени реакции системы слабо влияет на основные характеристики САЭ. Для таких программ на поле ДРП может быть выделен участок памяти, куда программы будут загружаться по схеме, аналогичной изобра-

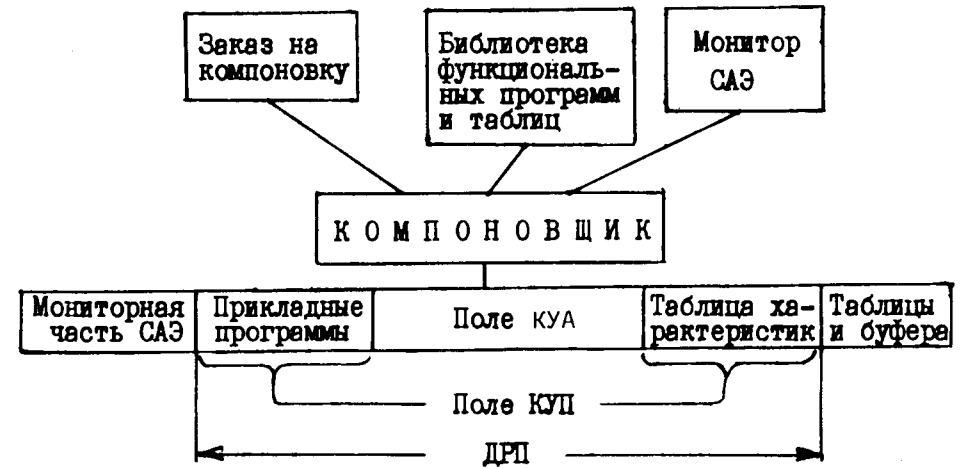


Рис.1. Упрощенная схема алгоритма компоновки.

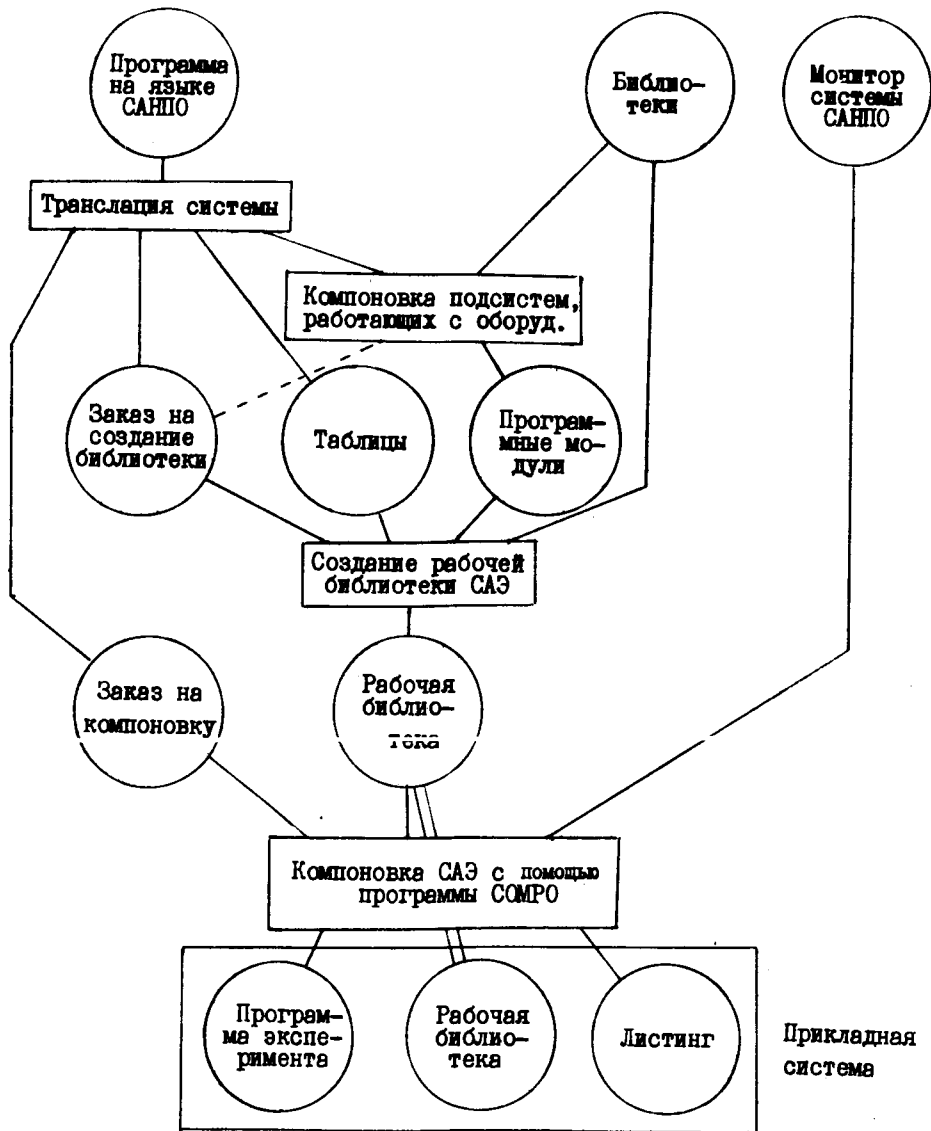


Рис.2. Положение компоновки в технологическом процессе создания САЭ.

женной на рис.1, но уже в процессе работы САЭ. Если на этом выделенном участке нет места для загрузки требуемой программы, выполняется очистка участка и повторная попытка загрузить эту программу. Такой алгоритм КУА, похожий в основном на алгоритм ИС-2^{10/}, реализован авторами^{11/} и обеспечивает построение автоматически управляемой конфигурации.

В следующем разделе описана программа-компоновщик, реализующая режим КУП при построении САЭ и обеспечивающая возможность дальнейшего функционирования САЭ в режиме КУА.

4. ПРОГРАММА-КОМПОНОВЩИК COMPO

4.1. Общая организация программы

Программа COMPO, блок-схема которой представлена на рис.3, написана на языке MACRO-11^{16/} и имеет оверлейную структуру с одним оверлейным полем. Каждый из оверлейных сегментов /их пять/ имеет свое функциональное назначение. Управляющий сегмент постоянно находится в памяти и содержит все общие переменные и подпрограммы, которые используются несколькими оверлейными сегментами.

Первый сегмент выполняет анализ строки исходных параметров, присвоение начальных значений переменным и буферам, проверку наличия в рабочей библиотеке модулей, перечисленных в заказе на компоновку. Для этого читается файл задания, структура которого имеет следующий вид:

Адрес слова	Содержание
0	Ресурс ОЗУ для САЭ /длина в байтах/
2	Дистанция от конца монитора до начала ДРП /область системных таблиц/
6	Указатель начала TX
12	Номер блока с информацией о перемещаемых адресах
с 14 по 116	Не используется
120	Начало поля записей /по 8 слов/ заказа на компоновку
....	Терминатор - запись с нулем во 2-м слове
....	Информация о перемещаемых адресах.

Основная часть файла - записи запросов на загрузку модулей. Содержание такой записи описывается следующей таблицей.

Адрес	Содержание
0	Тип запроса
2,4	Название модуля в коде RAD-50
6	Расширение названия в коде RAD-50
10	Начальный адрес внутри таблицы /номер блока/
12	Счетчик байтов, т.е. длина таблицы /или ноль/
14	Адрес загрузки /или ноль/
16	Ноль

В этой таблице тип запроса - одно слово, у которого значение четного байта определяет характер загружаемой информации, спо-

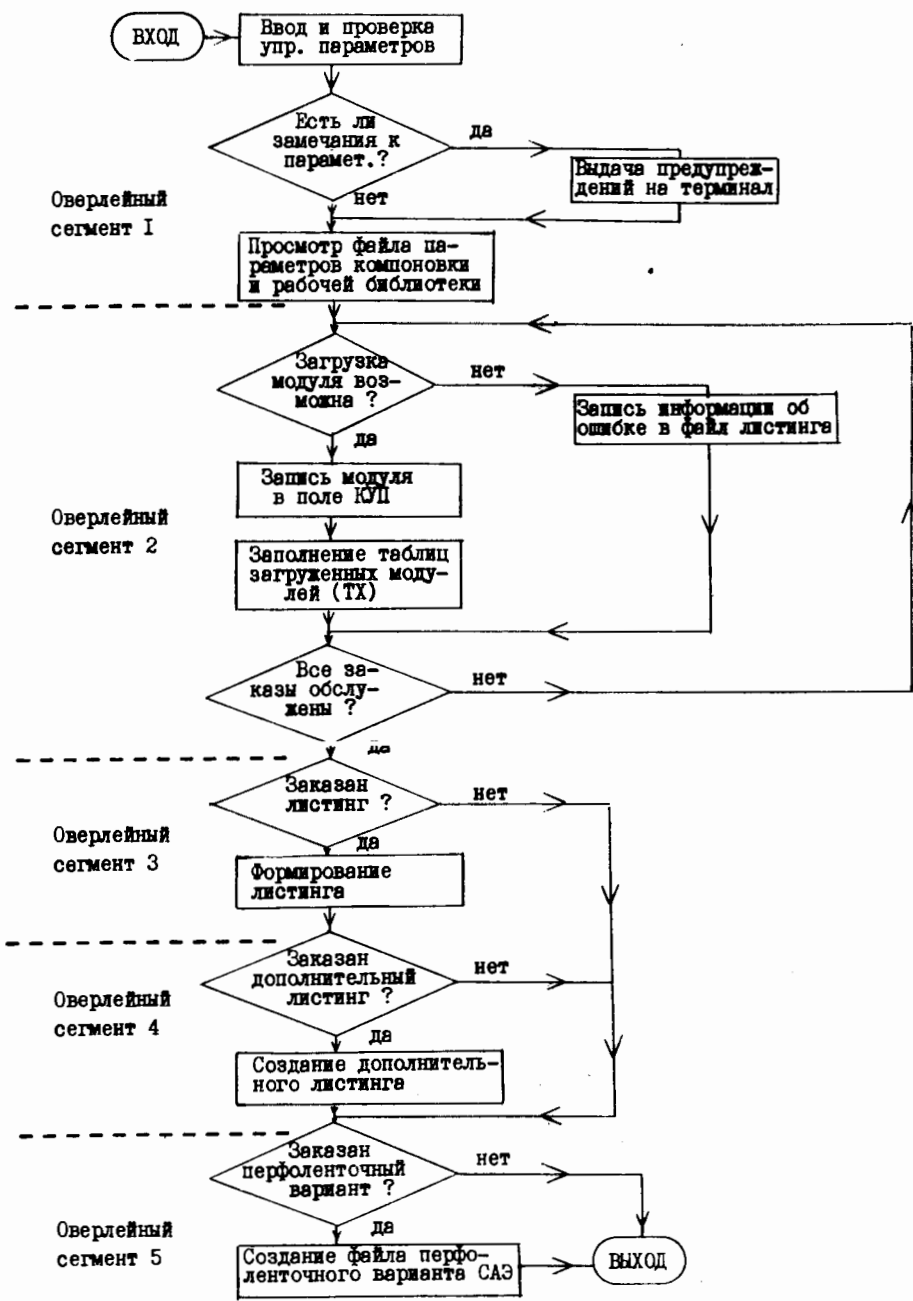


Рис. 3. Блок-схема программы-компоновщика COMPO.

соб загрузки и способ представления адреса загрузки. Если 7-й разряд типа запроса содержит 1, то указан абсолютный адрес загрузки, иначе - адрес относительно конца монитора. Характер информации описывается одним из четырех кодов:

- 0 - загрузка программы в формате .REL,
- 1 - загрузка таблицы,
- 3 - загрузка ненулевых слов таблицы,
- 4 - загрузка таблицы на поле ДРП.

В конце файла запросов находится информация о перемещаемых адресах, формат представления которой аналогичен принятому в операционной системе.

Второй оверлейный сегмент выполняет основную работу - компоновку САЭ. Он переписывает в начало файла образа системы монитор САНПО, загружает на поле программы указанные в задании программные модули, заполняет системные таблицы и буфера, настраивает перемещаемые адреса.

При записи компонентов в файл системы составляется карта занятых участков ОЗУ.

Третий и четвертый сегменты создают основной и дополнительный листинги компоновки.

Пятый сегмент содержит программу, выполняющую преобразование файла системы из формата .SAV в формат .LDA. Такое преобразование выполняется с учетом карты занятых участков с целью сократить объем работы при перфорации файла в формате .LDA^{1/6/}.

4.2. Управление программой

Общий вид командной строки следующий:

```
outfiles/switches=infiles/switches
```

Здесь infiles и outfiles - списки спецификаций исходных и результирующих файлов, switches - ключи, модифицирующие способ работы программы. В качестве исходных файлов указываются файлы задания компоновки системы, рабочей библиотеки и используемого монитора, в качестве результирующих - файл скомпонованной системы и листинга. Принимаемые по умолчанию названия файлов библиотеки и монитора - SANPO .LBR и SANPO.MON. Порядок файлов в командной строке подсказывается программой - компоновщиком после ее вызова.

Если опущено название файла задания, то компоновка не выполняется, а получается только листинг системы. Для отказа от листинга достаточно опустить спецификацию файла листинга. Принятые по умолчанию расширения названий файлов системы, листинга, задания, библиотеки, монитора соответственно -.SAV, .LST, .COM, .LBR, .MON. В качестве ключей могут использоваться следующие: /F - создать расширенную версию листинга системы; /L - создать файл системы в формате .LDA, предназначенном для загрузки системы не с диска, а с перфоленты.

Например, строка
 S,TT:/F=S,LIBR,MONI.SAV,
 означает: создать файл системы S.SAV на устройстве DK0, вывести расширенный (/F) листинг компоновки на терминал, задание на компоновку взять из файла DK0:S.COM, программные модули и таблицы взять из библиотеки DK0:LIBR,LBR, а монитор - из файла DK0:MONI.SAV.

4.3. Листинг компоновки

Пример листинга представлен на рис.4.

Первая строка листинга содержит название, номер версии программы COMPO и дату компоновки.

В следующих двух группах строк приведены спецификации исходных и результирующих файлов.

Далее могут появиться сообщения об ошибках, обнаруженных во время компоновки. Такие строки начинаются парой символов "Ж".

Заголовком FUNCTIONS начинается список модулей, успешно включенных в состав системы. Пронумерованные строки списка для каждого модуля содержат имя, адреса начала, конца и точки входа в модуль в теле системы, а также адрес элемента TX. Символ "S" в конце строки означает, что модуль включен в состав подсистемы /12/. Строки с таким символом содержат нули вместо адреса начала и конца модуля.

Следующая таблица подобна предыдущей, но включает в себя информацию о размещении элементов данных и таблиц. В конце строк ее могут присутствовать специальные символы:

Ж - из данной таблицы переписаны в систему только ненулевые элементы,

! - для размещения в системе данной таблицы компоновщику были указаны абсолютные адреса.

Такую /основную/ часть листинга завершают общие данные о системе в целом: длина системы, длина мониторной части, адреса начала DRP, начала поля АУК, начала TX, конца поля АУК.

Дополнение к листингу, печатаемое по ключу /F/, содержит в первой строке информацию о необходимых ресурсах памяти, характеристику используемой операционной системы и название созданной системы.

В следующих строках описаны используемые поля памяти /13/. Для каждого поля указан его номер, слово состояния, название устройства, граничные адреса и длина поля. Для файловых устройств вместо адресов приводится спецификация файла.

Далее в табличной форме представлена спецификация системы, дан полный перечень элементов данных /ЭД/ и ассоциированных с ними прикладных операций. Для каждого ЭД приводятся: его номер /в шапке таблицы помечен надписью NO/, название /NAME/, тип элемента данных /KOA/, спецификация данных /TYPE/, слово состояния /STATUS/, адрес описания ЭД в системных таблицах /SUBSC/

"SANPO" TASK COMPOSER VERSION V01.2 29-SEP-83

INPUT FILES:

PARAMETERS -DK : VEGA .COM
 LIBRARY -DK : SANPO .LBR
 MONITOR -DK : SANPO .MON

OUTPUT FILES:

TASK -DK : VEGA .SAV
 LIST -DK : VEGA .LST
 PARAMETERS -DK : PARAM .OUT

FUNCTIONS :

NO.	NAME-EXT	LDADDR	ENDADD	START	TXADDR
1.	TYCTRL.REL	13342	14744	13406	107154
2.	TYFUN.REL	14744	21036	15010	107136
3.	ERRRES.REL	21036	30166	21132	107120
4.	ODT.REL	30166	36056	30416	107102
5.	IORT.REL	36056	41036	36524	107064
6.	UPFLTR.REL	41036	42072	41074	107046
7.	SKPFLT.REL	42072	42566	42130	107030
8.	SUSPND.REL	42566	43040	42630	107012
9.	UNSET.REL	43040	43312	43102	106774
10.	TIMOUT.REL	43312	44020	43370	106756
11.	RUN.REL	44020	50434	44124	106740
12.	DND.REL	50434	53540	50554	106722
13.	TGENER.REL	0	0	53302	106704 S
14.	BKSENB.REL	0	0	53410	106666 S

DATA & ARRAYS :

NO.	NAME-EXT	LDADDR	ENDADDR	LENGTH	FR.BLK
1.	UPFLTB.BUF	113470	114000	310	0 #
2.	BFLIST.BUF	113160	113470	310	0 #
3.	VEGA.FT	1000	2000	1000	0 #!
4.	VEGA.DBT	12516	13342	624	0

** CHECK SUM ERROR IN:
 SET .REL

TASK'S LENGTH: 120000
 MONITOR'S END AT: 12516
 START OF DRP AT: 13342
 FIRST FREE ADDRESS ON DRP: 53540
 START OF TX AT: 107154
 FIRST FREE TX POINTER: 106650

CORE: 120000 STATUS: 0000000000000000 OPERATION SYSTEM: SJU EXPERIMENT: VEGA

FIELDS :

NO.	STATUS	DEV	STADDR	ENDADDR	LENGTH
1.	0000001010000001	RK0	EXPDAT.000		80. # 256. WORDS

GENERAL AREAS :

NO.	NAME	KOA	NOF	TYPE	STATUS	SUBSC	HIER	STADD.	LENGTH
									[VALUES]
1.	E	EV.			1001000000000000	12556			
	ACTION:	SET		("ERRRES")					
		MOVE		(ACT, 1)					
		DOWN		(E)					
2.	BUF1	BUF	0.	INT	1000000000000000	12604		114000	4000
	ACTION:	HISTOG		(BUF1,HIST)					
		IORT		("WRITE",BUF1,OUTB)					
3.	OUTB	BUF	1.	INT	1000000000000001	12632		0	120
	ACTION:	IORT		("CLOSE", 1)					
		SUSPND		("RUN")					
		UNSET		("RUN")					
		TIMOUT		()					
4.	DISP	EV.			0001000000000000	12660			
5.	DIALOG	EV.			0001000000000000	12706			
6.	UPFLTB	BUF	0.	INT	0000000000000000	12734		113470	310
7.	BFLIST	BUF	0.	INT	0000000000000000	12762		113160	310
8.	HIST	BUF	0.	INT	0000000000000000	13010		107160	4000
9.	ACT	VAR		INT	0000100000000000	13036		[1]	
10.	TIME	VAR		INT	0000100000000000	13064		[+0]	11

Рис.4. Листинг компоновки САЭ программой COMPO.

и номер другого ЭД, иерархически ^{/13/} связанного с данным /HIER/. Для буферов указан также номер поля /NOF/, на котором размещены данные, начальный адрес /STADD/ и длина /LENGTH/. Если переменным присвоены начальные значения (VALUES), то они приводятся в квадратных скобках.

После строк описания ЭД, с которыми ассоциированы какие-либо действия, приводится перечень вызываемых программ с указанием параметров. Такой список начинается словом ACTION.

Помимо листинга компоновки на терминале могут быть напечатаны сообщения, касающиеся в основном формальных ошибок оператора, или информация о состоянии программы.

ЗАКЛЮЧЕНИЕ

Разработаны алгоритмы и реализованы программы, выполняющие компоновку САЭ из компонентов, предварительно созданных и помещенных в рабочую библиотеку, путем извлечения из нее модулей в соответствии с описанием нужной конфигурации программного обеспечения. К существенным свойствам такого способа создания САЭ можно отнести следующие:

1/ Возможность комбинировать режим управления конфигурацией программного обеспечения со стороны пользователя и режим автоматической коррекции конфигурации в соответствии с изменением условий во время работы системы. Такая возможность обусловлена тем, что при компоновке, управляемой пользователем, и при динамическом изменении состава программ в оперативной памяти во время работы САЭ используются модули одинакового формата, размещаемые в одной рабочей библиотеке ^{/8,9/}. Это свойство существенно облегчает модификацию САЭ, т.к. дает возможность заменять отдельные функциональные программы после компоновки САЭ путем замены их версий в рабочей библиотеке.

2/ Повышение уровня преемственности разработок благодаря имеющейся возможности изменять в созданной системе отдельные компоненты /например, функциональные программы/, не затрагивая остальной части САЭ.

Отметим, что таким способом обеспечивается в некоторой мере преемственность комплексной отладки.

3/ Принятая в настоящее время организация способствует интеграции опыта освоения проблемной области, т.к. функциональные возможности, необходимость которых под вопросом, а полезность бесспорна, обычно не включаются в САЭ, даже если есть свободные ресурсы ОЗУ. При описанной организации такие программы могут включаться в рабочую библиотеку, на что не расходуется оперативная память или дополнительное время разработчика.

4/ На предварительных этапах создания компонентов САЭ максимально используются имеющиеся в ОС средства программирования /трансляторы, редактор связей и др./, что сократило объем дополнительных разработок при выполнении данной работы.

Основной целью принятого подхода является сокращение затрат на создание конкретных САЭ.

Отличительными особенностями разработанного подхода и реализованных программ являются:

1. Возможность изменить конфигурацию программного обеспечения после компоновки систем путем замены пользователем функциональных программ в рабочей библиотеке.

2. Возможность использовать созданные автором ранее ^{/11/} и автоматически работающие средства изменения конфигурации САЭ.

3. В качестве динамических компонентов САЭ приняты модули в перемещаемом формате загрузки ^{/8,9/}.

Программа-компоновщик эксплуатируется в ЛНФ ОИАИ ^{/14-18/}, ИАиЗ г.Новосибирска ^{/19/} и некоторых других организациях. Опыт эксплуатации показывает полезность использования основных принципов данной разработки.

В заключение автор пользуется возможностью поблагодарить И.М.Саламатина за руководство разработкой и своих коллег за помощь в работе.

ЛИТЕРАТУРА

1. Балука Г. и др. Р10-12960, Дубна, 1980.
2. Колпаков И.Ф. Автометрия №3, 1980, с. 28-33.
3. Wirth N., MODULA-2. Institut für Informatik der ETH Zurich, Nr. 36, 1980.
4. Йенсен К., Вирт Н. Паскаль. Руководство для пользователя и описание языка. "Финансы и статистика", М., 1982.
5. Язык программирования АДА /предварительное описание/. "Финансы и статистика", М., 1981.
6. Силинов Е.М., Семик В.П. Программное обеспечение УВК СМЗ. Приборы и системы управления, 1977, №10, с. 15-17.
7. Дисковая операционная система реального времени ДОС РВ мини-ЭВМ МЕРА СЛМАС 125/М 4А, Варшава, 1982.
8. Балука Г., Саламатин И.М., Хрыкин А.С. ОИАИ, Р10-12546, Дубна, 1979.
9. Балука Г., Саламатин И.М., Хрыкин А.С. ОИАИ, Р10-12545, Дубна, 1980.
10. Шура-Бура М.Р. Интерпретирующая система для М-20, ВЦ АН СССР, М., 1965.
11. Балука Г., Островной А.И. ОИАИ, Р10-13004, Дубна, 1980.
12. Балука Г. и др. ОИАИ, Р10-80743, Дубна, 1980.
13. Островной А.И., Саламатин И.М. ОИАИ, Р10-80-423, Дубна, 1980.

14. Балагуров А.М. и др. ОИЯИ, P10-80-824, Дубна, 1980.
15. Балука Г. и др. ОИЯИ, P10-80-825, Дубна, 1980.
16. Вагов В.А. и др. ОИЯИ, P10-80-826, Дубна, 1980.
17. Бакалов Т. и др. ОИЯИ, P10-82-522, Дубна, 1982.
18. Гриднев Г.Ф., Саламатина Т.С. ОИЯИ, 10-83-598, Дубна, 1983.
19. Саламатин И.М. и др. Автометрия, 1981, №4, с. 60-69.

НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

Вы можете получить по почте перечисленные ниже книги, если они не были заказаны ранее.

	Труды VI Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1978 /2 тома/	7 р. 40 к.
	Труды VII Всесоюзного совещания по ускорителям заряженных частиц, Дубна, 1980 /2 тома/	8 р. 00 к.
D11-80-13	Труды рабочего совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике, Дубна, 1979	3 р. 50 к.
D4-80-271	Труды Международной конференции по проблемам нескольких тел в ядерной физике. Дубна, 1979.	3 р. 00 к.
D4-80-385	Труды Международной школы по структуре ядра. Алушта, 1980.	5 р. 00 к.
D2-81-543	Труды VI Международного совещания по проблемам квантовой теории поля. Алушта, 1981	2 р. 50 к.
D10,11-81-622	Труды Международного совещания по проблемам математического моделирования в ядерно-физических исследованиях. Дубна, 1980	2 р. 50 к.
D1,2-81-728	Труды VI Международного семинара по проблемам физики высоких энергий. Дубна, 1981.	3 р. 60 к.
D17-81-758	Труды II Международного симпозиума по избранным проблемам статистической механики. Дубна, 1981.	5 р. 40 к.
D1,2-82-27	Труды Международного симпозиума по поляризационным явлениям в физике высоких энергий. Дубна, 1981	3 р. 20 к.
P18-82-117	Труды IV совещания по использованию новых ядерно-физических методов для решения научно-технических и народнохозяйственных задач. Дубна, 1981.	3 р. 80 к.
D2-82-568	Труды совещания по исследованиям в области релятивистской ядерной физики. Дубна, 1982.	1 р. 75 к.
D9-82-664	Труды совещания по коллективным методам ускорения. Дубна, 1982.	3 р. 30 к.
D3,4-82-704	Труды IV Международной школы по нейтронной физике. Дубна, 1982.	5 р. 00 к.
D2,4-83-179	Труды XV Международной школы молодых ученых по физике высоких энергий. Дубна, 1982.	4 р. 80 к.
	Труды УШ Всесоюзного совещания по ускорителям заряженных частиц. Протвино, 1982 /2 тома/	11 р. 40 к.
D11-83-511	Труды совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике. Дубна, 1982.	2 р. 50 к.
D7-83-644	Труды Международной школы-семинара по физике тяжелых ионов. Алушта, 1983.	6 р. 55 к.
D2,13-83-689	Труды рабочего совещания по проблемам излучения и детектирования гравитационных волн. Дубна, 1983.	2 р. 00 к.

Рукопись поступила в издательский отдел
4 ноября 1983 года.

Заказы на упомянутые книги могут быть направлены по адресу:
101000 Москва, Главпочтамт, п/я 79
Издательский отдел Объединенного института ядерных исследований

**ТЕМАТИЧЕСКИЕ КАТЕГОРИИ ПУБЛИКАЦИЙ
ОБЪЕДИНЕННОГО ИНСТИТУТА ЯДЕРНЫХ
ИССЛЕДОВАНИЙ**

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

Балука Г. 10-83-761
Программа-компоновщик систем автоматизации эксперимента в комплексе САНПО

Обсуждаются принципы, позволяющие сократить затраты на создание серии систем автоматизации эксперимента /САЭ/ путем применения автоматизированного способа компоновки программного обеспечения САЭ из функциональных программ. Описаны алгоритмы и реализующая их программа. Отличительной особенностью построенной описанным способом САЭ является то, что она позволяет производить замену версий и введение новых функциональных программ без редактирования текста всей системы.

Работа выполнена в Лаборатории нейтронной физики ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1983

Baluka G. 10-83-761
Program-Linker for the Experiment Automation in the SANPO System

The concepts of generation of a series of the experiment automation systems (EAS) with a minimum human's resource consuming are discussed. This is due to the application of automatized way of compiling the software of EAS from functional programs. Algorithm and their program are described. The main peculiarity of the concept is the possibility to substitute versions and to add new functional programs without editing text of the whole system.

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

Communication of the Joint Institute for Nuclear Research. Dubna 1983

Перевод О.С.Виноградовой