

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА



Ц8406

H-789

1459/2-77

П.Нойберт

18/10-77

11 - 10279

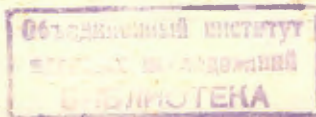
О РЕАЛИЗАЦИИ
ПРЕДВАРИТЕЛЬНОГО КОМПИЛЯТОРА
ДЛЯ ЯЗЫКА КАМАК
В ОПЕРАЦИОННОЙ СИСТЕМЕ ЭВМ НР 2116 С

1977

11 - 10279

П.Нойберт

О РЕАЛИЗАЦИИ
ПРЕДВАРИТЕЛЬНОГО КОМПИЛЯТОРА
ДЛЯ ЯЗЫКА КАМАК
В ОПЕРАЦИОННОЙ СИСТЕМЕ ЭВМ **НР 2116 С**



Нойберг П.

11 - 10279

О реализации предварительного компилятора для языка КАМАК в операционной системе ЭВМ HP-2116 С

В работе сообщается о реализации компилятора для предложенного комитетом ESONE языка КАМАК. Соответствующие КАМАК-команды расширяют язык ФОРТРАН. Таким образом, создание программ для управления измерительной аппаратурой и обработки данных выполняется на привычном для экспериментатора алгоритмическом языке. Компилятор работает в дисковой операционной системе.

Работа выполнена в Лаборатории ядерных проблем ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1977

Быстрое развитие вычислительной техники привело к тому, что ЭВМ стали использоваться для решения не только чисто математических задач, но и задач управления и регулирования. Это потребовало соединения ЭВМ с различными приборами и устройствами, дающими возможность принимать и выдавать соответствующие данные.

Различные варианты соединения обусловили ясное и четкое определение свойств этих мест соединений (interface). На основе требований экспериментальной техники ядерной физики был создан стандарт КАМАК^{1,2/}. Команды, посылаемые в модуль КАМАК со стороны управляющей ЭВМ, должны указывать, что ($\rightarrow \bar{F}$) и где ($\rightarrow B, C, N, A$) надо выполнить. Это делает их похожими на собственные команды ЭВМ. Со своей стороны, модуль может посылать требования на обслуживание центральным управлением ($\rightarrow L$).

Ввиду сложности управляемых процессов целесообразно поручить ЭВМ организовать работу устройств в стандарте КАМАК. Это значит, что в программе для ЭВМ определяется последовательность необходимых команд управления и команд обмена данными.

1. ЯЗЫК ПРОГРАММИРОВАНИЯ КАМАК

Комитет ESONE разработал предложение о создании языка для программного обеспечения оборудования в стандарте КАМАК^{1,3/}.

1.1. Предпосылки создания языка КАМАК

Предложение комитета ESONE обусловлено тем, что в программе для обслуживания устройств в стандарте КАМАК должны быть не только операторы обмена данными и операторы управления процессом, но и операторы для обработки данных.

Известные языки высокого уровня (например: ФОРТРАН, ПЛ/1) в основном дают возможность обрабатывать данные. Поэтому было предложено использовать эти языки, расширив их введением дополнительных операторов. Расширение может осуществляться следующим образом:

- отдельные участки программы или макрокоманды могут быть написаны на ассемблере;
- возможно обращение к подпрограммам, которые могут быть использованы при программировании на языках высокого уровня;
- могут быть привлечены дополнительные вспомогательные средства, которые включаются в систему ввода-вывода.

Недостаток первого варианта состоит в том, что требует от программиста хорошего знания внутренней структуры ЭВМ и контроллера. Это замечание в полной мере относится также к языку ИМЛ (IML) ^{4/}.

Ввиду распространенности языков высокого уровня целесообразно программировать управление оборудованием в стандарте КАМАК на уровне этих языков. Возможно либо обращение к подпрограммам, которые вызываются на этих языках, либо использование языка КАМАК, который является языком столь же высокого уровня. Эти два подхода второго варианта обеспечивают удобное и простое программирование. Реализация третьего варианта связана со значительными затратами.

1.2. Структура программы на языке КАМАК

Программа КАМАК содержит наряду с операторами взаимодействия с устройствами в стандарте КАМАК,

```
non-CAMAC-text
CAMACSEGMENT
CEQV      equivalence-statements
CNAME    CAMAC-naming-statements
CDMD     demand-naming-statements
CDCL     software-naming-statements
CREF     external-reference-statements
CACT     CAMAC-action-statements
CAMACSEGMENTEND
non-CAMAC-text
```

Рис. 1. Структура КАМАК - программы.

которые должны объединяться в так называемом КАМАК-сегменте, также полный комплект операторов одного из известных языков высокого уровня. Один КАМАК-сегмент может состоять только из КАМАК - операторов или может быть дополнен операторами других языков с помощью ключевых слов "ENDCAMAC" и "BEGINCAMAC".

Основная структура одного КАМАК - сегмента показана на рис. 1. КАМАК-операторы будут в дальнейшем кратко описаны.

Операторы CEQV, CNAME, CDMD, CDCL и CREF образуют декларативную часть сегмента.

1.2.1. Операторы эквивалентности (equivalence statements)

Имеется два вида операторов эквивалентности.

Первый из них дает возможность присваивать названия существующим символам или менять их. Например, в программу вместо функционального кода F(27) можно ввести название "CHECK".

Второй дает возможность вводить символьные константы и присваивать им определенные значения. Они действуют только во время трансляций.

1.2.2. Операторы наименования оборудования в стандарте КАМАК (CAMAC naming statements)

Эти операторы дают возможность вводить наименования вместо цифровых ссылок для величин В, С, N, А и I, например, В(1)С(3)N(1)A(0).

При помощи одного имени можно выбрать отдельный элемент или совокупность элементов, объединенных в одном адресном поле. Эти возможности показаны на следующих примерах.

1. ICC = C(1)N(1)A(0).

Запрашивается счетчик, который находится в первом крейте в модуле на станции 10 в подадресе А(0).

2. ICC(1:20) = C(1)N(1:10:2)A(0:3).

Запрашивается крейт с несколькими счетчиками. Каждый из них занимает один подадрес из диапазона А(0)... А(3). Тетрада счетчиков находится в одном из пяти модулей двойной ширины, которые занимают станции N(1), N(3), N(5), N(7) и N(9). Таким образом, возникло поле адресов КАМАК - оборудования. Оператор ICC(1:20) = C(1)N(1,3,5,7,9)A(0,1,2,3)

эквивалентен описанному выше.

3. IBRA = B(1)

NCR = IBRA C(1)

KMOD1 = NCR N(1).

В правой части этого ряда операторов находятся символические названия, которые уже были введены ранее и используются при образовании новых адресов. Так можно строить иерархически организованные адресные структуры.

1.2.3. Операторы наименования источников требований (demand naming statements)

Вышеупомянутый сигнал L может быть определен с помощью оператора наименования источника требования. В этот оператор входят

- название модуля, уже приведенное в 1.2.2, служащее для обозначения места источника требования,
- номер разряда источника требования в слове "graded LAM-word" контроллера крейта. Например:

LAM1 = KMOD1 GL 12.

Таким образом осуществляется простая связь между сигналами L, идущими от модулей в стандарте КАМАК, и программным обеспечением.

1.2.4. Операторы наименования для программного обеспечения (software naming statements)

Такие операторы необходимы, чтобы указать область данных для обмена, которая используется КАМАК-сегментом.

Они обеспечивают:

- связь между КАМАК - сегментом и остальной частью программ / "non-CAMAC-text", сравни рис. 1/.
- определение длины слов /COMLEN $\hat{=}$ 16 разрядов, CAMLEN $\hat{=}$ 24 разрядов/.
- возможность транслятору проводить различные тесты во время трансляции.

1.2.5. Операторы внешних обращений (external reference statements)

Эти операторы позволяют использовать во всех сегментах одной программы названия, которые были уже определены с помощью вышеприведенных правил. Это особенно удобно при большом объеме КАМАК-оборудования, так как в этом случае операторы наименования отдельных элементов могут быть записаны в программе

только один раз. В других сегментах их можно использовать с помощью внешних вызовов.

1.2.6. Операторы выполнения (CAMAC action statements)

Они служат для общей связи между ЭВМ и оборудованием в стандарте КАМАК. Существуют следующие 6 типов этих операторов:

1. Операторы ввода данных: READ ext-ref int-ref:
ext-ref определяет название одного модуля или название целого поля соответственно главе 1.2.2. или дает так называемую непосредственную ссылку /direct-reference/, которая содержит необходимые величины для В, С, N, А и I.
int-ref определяет в памяти массив, предусмотренный для приема данных /см.1.2.4/.
2. Операторы вывода данных: WRITE int-ref ext-ref.
ext-ref имеет те же значения, что и операторы ввода.
int-ref определяет массив в памяти, предусмотренный для вывода данных.

3. Операторы передачи данных между КАМАК - устройствами: TRANSFER ext-ref1 ext-ref2.
Первая внешняя ссылка обозначает источник, вторая - приемник данных внутри КАМАК - устройств.

4. Операторы управления обозначают различные функции в КАМАК-оборудовании, для выполнения которых не требуются никакие внутренние ссылки /например: сброс, выключить, включить/.

Например:

CLEAR ext-ref

Нужно заметить, что в операторах 1,...,4 вместо названия операции могут стоять:

- все КАМАК - функции в форме F(i), где $i = \emptyset, 1, 2, \dots, 31$;
- или названия различных функций, которые были введены операторами параграфа 1.2.1.

5. Операторы условного перехода. Эти операторы разрешают разветвления в программах в зависимости

от определенных состояний оборудования в стандарте КАМАК. Например, в случае

IF LAM ext-ref GOTO 120

происходит передача управления программой оператору с меткой 120, если в модуле, обозначенном через внешнюю ссылку (ext-ref), установлен сигнал L. Если сигнал L не установлен, выполняется команда оператора, следующего за оператором IF.

6. Кроме этого, существуют некоторые операторы, которые обеспечивают взаимодействие между КАМАК-сегментом и операционной системой.

1.3. Проблема включения языка КАМАК в язык высокого уровня

Как уже было сказано в параграфе 1.1., было предложено операторы для обслуживания устройств в стандарте КАМАК включить в подходящий язык высокого уровня. Это включение может осуществляться по-разному, в зависимости от этого получают различные варианты реализации. Если полностью смешать все КАМАК-операторы с операторами другого языка, то потребуется такой компилятор, который сможет транслировать весь новообразованный язык целиком. Рис. 2 показывает возможность такого варианта.

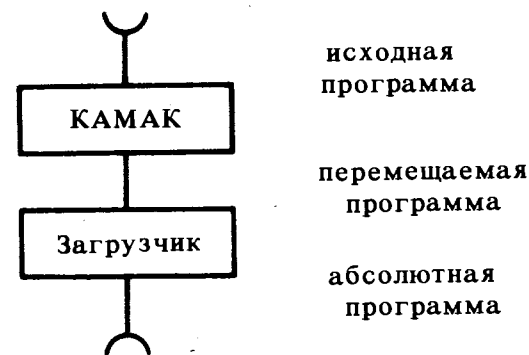


Рис. 2. КАМАК - компилятор, вариант 1.

Реализация такого варианта связана с большими затратами, так что исследуются другие возможности. Варианты, в которых КАМАК-операторы добавляются к языку ФОРТРАН, причем КАМАК - операторы объединяются в одном КАМАК-сегменте, показаны на рис. 3 и 4.

Реализация обоих вариантов связана с определенными трудностями. Вариант, показанный на рис. 3, несколько удобнее, так как после компилирования сразу получается программа в перемещаемом формате. Но он требует определенных связей между программами-компиляторами, особенно в отношении распределения памяти, ведения списков и т.д.

Если во время разработки КАМАК-компилятора изменения в ФОРТРАН-компиляторе недопустимы, тогда этот вариант практически невозможно реализовать. На рис. 4 изображен другой вариант. Необходимо, чтобы операторы программы, генерированные КАМАК-компилятором, соответствовали синтаксическим правилам языка, в который включен КАМАК-язык /например, язык

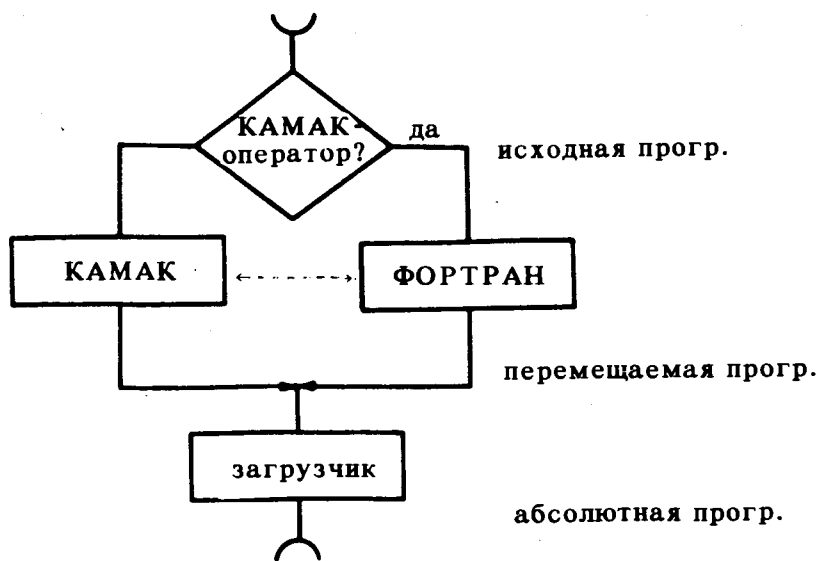


Рис. 3. КАМАК - компилятор, вариант 2.

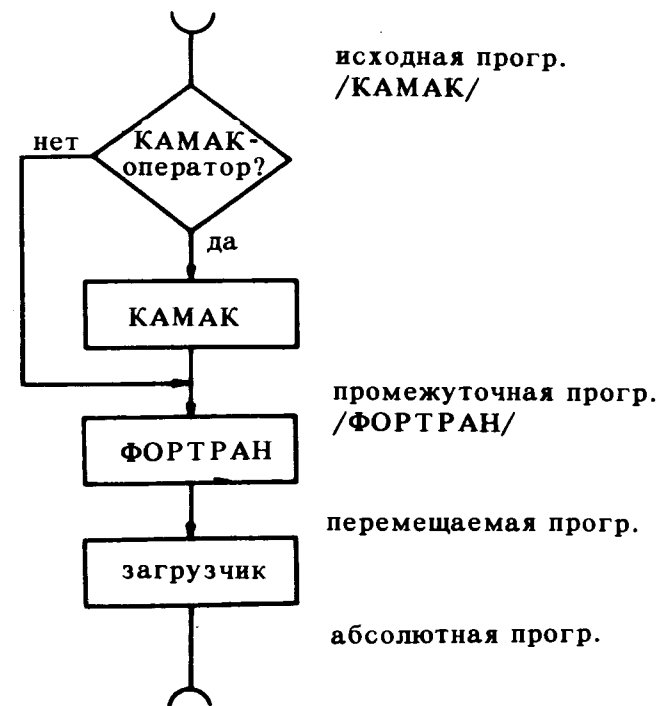


Рис. 4. КАМАК - компилятор, вариант 3.

ФОРТРАН/. Это требование отражается на эффективности компилированной программы. Затраты на изменение соответствующих модулей операционной системы незначительны.

II. РЕАЛИЗАЦИЯ КОМПИЛЯТОРА ДЛЯ ЯЗЫКА КАМАК

II.1. Краткое описание операционной системы ДОС

Здесь коротко рассмотрены только проблемы, связанные с компиляцией исходных программ.

Для гибкой организации часто повторяющихся процессов, таких, как компиляция, загрузка и выполнение программ, операционная система имеет в наличии библиотеку, в которой могут накапливаться программы, находящиеся на разных стадиях разработки. Особенно важно, что перевод программ первоначальной формы в машинный код происходит в один проход, в частности, без применения таких внешних накопителей, как бумажная лента.

Нам надо было разработать такой способ реализации КАМАК-компилятора, который бы обеспечил эффективную работу системы и обращение к ней в принятой форме.

Очень важно, что в предложенном способе реализации компилятора имеется возможность вызывать и вводить в транслированную программу подпрограммы, написанные на другом языке.

Обычно компилятор при трансляции исходных операторов генерирует последовательность соответствующих машинных команд. Для КАМАК - компилятора, изображенного на рис. 4, такой возможности нет. Команды, генерированные этим компилятором, должны соответствовать синтаксическим правилам компилятора, работающего вслед за КАМАК-компилятором. Поэтому машинные команды, которые реализуют все КАМАК-операторы, объединены в библиотеку подпрограмм. КАМАК-компилятор при трансляции КАМАК-операторов должен постоянно генерировать вызовы с параметрами соответствующих подпрограмм.

II.2. Принцип реализации

Из предшествующих разъяснений видно, что язык КАМАК удобно включить в язык высокого уровня ФОРТРАН операционной системы ЭВМ ИР 2116. ФОРТРАН широко распространен, и поэтому обеспечивает большие возможности и удобство при программировании для применения КАМАК-оборудования.

II.2.1. Работа КАМАК - компилятора в операционной системе ДОС

Транслятор, схема реализации которого изображена на рис. 4, обрабатывает исходную программу, вводимую из библиотеки пользователя или с устройства ввода, выдавая при этом "чистую" ФОРТРАН-программу. После этого следует автоматический вызов ФОРТРАН - компилятора, который преобразует эту промежуточную программу в программу в перемещаемом формате. Компилятор вызывается с помощью следующей последовательности:

:PROG,CMC,p1,p2,p3,p4,p5.

где

- p1 : логический номер устройства для ввода исходной программы,
- p2 : логический номер АЦПУ,
- p3 : логический номер перфоратора,
- p4 : число строк в странице для p2.
- p5 : операция для загрузки и запуска.

КАМАК-компилятор проверяет и использует эти параметры и передает их в конце работы ФОРТРАН-компилятору. Компилятор состоит из 5 сегментов, которые выполняют следующие задачи:

Главный сегмент

- оценку заданных параметров,
- подготовку вызова различных сегментов,
- загрузку основных подпрограмм;

1 - й сегмент

- ввод исходной программы с перфоленты;

2 - й сегмент

- синтаксический анализ КАМАК-сегмента,
- построение списков для операторов эквивалентности, операторов наименования оборудования в стандарте КАМАК, операторов наименования источников требования,

операторов наименования для программного обеспечения,

- оптимизацию списков;

3-й сегмент

- трансляцию КАМАК-сегмента, т.е. генерирование необходимых операторов на языке ФОРТРАН,
- если требуется, печать исходной программы,
- если требуется, перфорирование транслированной программы,
- передачу транслированной программы на диск;

4-й сегмент

- поиск соответствующих начальных параметров для ФОРТРАН-компилятора,
- передачу заданных параметров ФОРТРАН-компилятору,
- вызов ФОРТРАН-компилятора.

Печать и перфорирование программы могут быть потребованы введением следующих параметров в оператор управления компилятора:

где $CMC[4][,LC][,LF][,LA][,F][,B]$

CMC: запуск компилятора ФОРТРАН-2 /после окончания работы КАМАК-компилятора/; это означает, что включение КАМАК-языка происходит в язык ФОРТРАН-2;

CMC4: все то же самое, что для языка ФОРТРАН-2;

LC: печать исходной программы на языке КАМАК;

LF: печать промежуточной ФОРТРАН-программы;

LA: печать промежуточной АССЕМБЛЕР-программы;

F: перфорирование промежуточной ФОРТРАН-программы;

B: перфорирование двоичной программы.

II.2.2. Библиотека КАМАК-подпрограмм - важная составная часть КАМАК-компилятора

Компилятор работает таким образом, что при трансляции КАМАК-сегмента он не генерирует последова-

тельность машинных команд, а использует с помощью оператора в ФОРТРАНе соответствующую последовательность машинных команд, которые находятся в форме подпрограмм в этой библиотеке. Благодаря этому, транслированные элементы соответствуют ФОРТРАН-синтаксису и создается оптимальная по гибкости и эффективности программа.

На рис. 5 перечислены соответствующие подпрограммы, которые гарантируют выполнение команд отдельных КАМАК-операторов.

Параметры, приведенные на рис. 5, имеют следующие значения:

ICMCC	- название главной COMMON -области для обработки требований,
ICMCS	- название COMMON -области для спецификации внутренних ссылок,
I1	- коды некоторых символов языка,
I2	- функциональные коды в стандарте КАМАК,
IEX1	- название внешних ссылок,
IEX2	- начальный элемент внешних ссылок,
IEX3	- конечный элемент внешних ссылок,
IINT1	- название внутренних ссылок,
IINT2	- начальный элемент внутренних ссылок,
IINTS	- соответствующий элемент в области ICMCS.

II.3. Отличие описываемого КАМАК-языка от предложенного комитетом

Предложенный комитетом ESONE язык КАМАК не может быть полностью реализован на ЭВМ HP 2116 из-за особенностей операционной системы этой ЭВМ. В системе ДОС ЭВМ HP 2116 невозможна параллельная работа нескольких программ, поэтому в нашем языке отсутствует режим "нескольких задач" (multi tasking).

Следующее отличие состоит в том, что в одной программе пользователя допускается только один КАМАК-сегмент. Это не накладывает больших ограничений на программиста при создании программы. Операционная система ДОС допускает сегментирование программ поль-

Оператор	Вызов подпрограммы
LINK (.....)	CALL CMC#3 (ICMCC(I))
ENDLAMROUT	CALL CMC#4
INITIALISE(...)	CALL CMC#5 (ICMCS(I))
IGNORE (.....)	CALL CMC#6 (ICMCC(I))
REPEAT (...)	CALL CMC#7 (...)
IF ...	CALL CMC1# (I1, IEX1, IEX2) GO TO ...
IFNOT ...	CALL CMC11 (I1, IEX1, IEX2) GO TO ...
P (...)... CLEAR ... CLEARLAM ... ENABLE ... SET ... DISABLE ... EXECUTE ...	CALL CMC2# (I2, IEX1, IEX2, IEX3)
INITIALISE ... SETINHIBIT ... CLEARINHIBIT ... ENABLEINT ... DISABLEINT ... CLEARSYS ...	CALL CMC21 (I2, IEX1, IEX2, IEX3)
P (...)... READ ... READCLR ... READCOMP...	CALL CMC3# (I2, IEX1, IEX2, IEX3, IINT1, IINT2, IINTS)
P(#)...EXIT... READ...EXIT...	CALL CMC31 (I2, IEX1, IEX2, IINT1, IINT2, GO TO ... IINTS)
P(...)... WRITE SETSEL CLEARSEL	CALL CMC32 (I2, IINT1, IINT2, IINTS, IEX1, IEX2, IEX3)
P(16)...EXIT... WRITE...EXIT...	CALL CMC33 (I2, IINT1, IINT2, IINTS, IEX1, IEX2, IEX3) GO TO
TRANSFER	CALL CMC 34 (I2, IEX1, IEX2, IEX3, I EX 1, I EX 2, I EX 3)
READLAM... READSTAT...	CALL CMC35 (I2, IEX1, IEX2, IEX3, IINT1, IINT2, IINTS)

Рис. 5. Библиотека подпрограмм.

зователя. Предлагаемый вариант языка КАМАК позволяет иметь в каждом сегменте программы один КАМАК-сегмент. С помощью аппарата внешних обращений организована связь между отдельными КАМАК-сегментами. Благодаря этому, пользователь получает удобную возможность обращаться к многочисленным КАМАК-блокам из разных мест своей программы.

III. ЗАКЛЮЧЕНИЕ

В этом сообщении описана реализация КАМАК-языка для ЭВМ HP 2116 на основе предложения ESONE - комитета. Был разработан язык для программирования управления КАМАК - оборудованием. Разработанный для операционной системы ДОС вариант может быть использован в системах BCS и RTE.

В заключение автор выражает благодарность А.Н.Синаеву, С.В.Медведю, И.Н.Чурину за полезное обсуждение, а также Н.Н.Альбрехту и Н.Г.Клюкину за помощь в подготовке рукописи.

ЛИТЕРАТУРА

1. CAMAC. A Modular Instrumentation System for Data Handling. ESONE Committee EUR 4100, 1972.
2. CAMAC. Organisation of Multy-Crate Systems. ESONE Committee EUR 4600, 1972.
3. CAMAC. Proposal for a CAMAC Language. CAMAC-Bulletin No. 5, Nov. 1972.
4. CAMAC. The Definition of IML (A Language for Use in CAMAC-Systems). Proposal of Software Working Group of ESONE 1974.

Рукопись поступила в издательский отдел
10 января 1977 года.