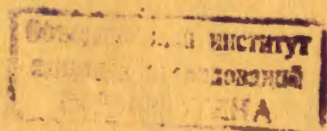


сообщения  
объединенного  
института  
ядерных  
исследований  
дубна



11-83-399

В.А.Загинайко

АЛГОРИТМЫ И СТРУКТУРЫ  
ДАННЫХ ТРАНСЛЯТОРА С ЯЗЫКА ФОРТРАН  
ДЛЯ ЭВМ С МАЛЫМ ОБЪЕМОМ  
ОПЕРАТИВНОЙ ПАМЯТИ

1983

Рассматриваемый в настоящей статье транслятор с языка Фортран работал в составе компилирующей системы БЭСМ-4<sup>/1/</sup> и давал возможность включать в библиотеку системы программы, написанные на Фортране, в варианте, близком к версии языка, реализованном на БЭСМ-6 в мониторной системе "Дубна".

Расширение возможностей входного языка этого транслятора по сравнению с вариантом транслятора 1971 года<sup>/2/</sup> потребовало изменения алгоритмов основных его блоков, так что в данном случае речь идет о совершенно новом трансляторе.

В рассматриваемом варианте транслятора сокращено количество информационных таблиц и изменена их структура, что позволило более эффективно использовать память ЭВМ в процессе трансляции и ослабить ограничения, накладываемые на размеры транслируемой программы, количество идентификаторов в ней и т.п.

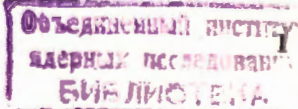
Программирование данного варианта транслятора, равно как и других компонент компилирующей системы БЭСМ-4<sup>/1/</sup>, выполнено машинно-независимым образом на языке "МАКРОС"<sup>/3/</sup>, что дает возможность перетрансляции компонент системы на ЭВМ с другой системой команд. Кроме того, путем замены макроопределений макроассемблера, входящего в состав транслятора и компилирующей системы, можно менять систему команд программы, получаемой в результате трансляции.

## 1. СТРУКТУРА И ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ТРАНСЛЯТОРА

Транслятор версии 1975 года в целом сохранил структуру транслятора версии 1971 года, описанного в работе<sup>/2/</sup>, однако в целях более эффективного использования памяти ЭВМ БЭСМ-4 в нем увеличено количество сегментов транслятора, т.е. частей программы, работающих в разное время на одном и том же участке оперативной памяти ЭВМ.

Транслятор включает в себя следующие сегменты:

- 1/ головной;
- 2/ основной;
- 3/ декларативный процессор,
- 4/ END-процессор,
- 5/ сегмент макроопределений,
- 6/ сегмент диагностики и отладки транслятора,
- 7/ сегмент обработки операторов-функций,
- 8/ сегмент обработки чисел с двойной точностью.



Структура транслятора, отражающая подчиненность блоков /т.е. возможность обращения к подпрограммам, входящим в подчиненный блок/, имеет следующий вид:

1. Головной блок транслятора.
  - 1.1. Блок чтения оператора.
  - 1.2. Блок опознавания оператора.
    - 1.2.1. Блоки декларативного процессора.
      - 1.2.1.1. Лексический анализатор и блок записи переменных и констант в таблицу.
      - 1.2.2. Блоки обработки неарифметических операторов.
        - 1.2.2.1. /=1.2.1.1./.
      - 1.2.3. Блоки обработки арифметических операторов.
        - 1.2.3.1. Арифметический блок.
          - 1.2.3.1.1. /=1.2.2.1./.
    - 1.3. Программирующий блок.
      - 1.3.1. Макроассемблер.

Головной сегмент включает в себя блоки 1, 1.1, 1.2.  
Основной сегмент включает в себя блоки 1.2.1.1, 1.2.2, 1.2.3, 1.2.3.1.

Декларативный процессор включает в себя блоки 1.2.1.

END-процессор содержит блоки 1.3 и 1.3.1.

В процессе работы транслятором формируются следующие рабочие поля /магазины, таблицы/:

- 1/ таблица величин /переменных и констант транслируемой программы/,
- 2/ таблица макрокоманд транслируемой программы /программирующая таблица в терминах <sup>13</sup>/,
- 3/ поле текста транслируемого оператора языка фортран /в коде ГОСТ/,
- 4/ таблица оператора /получаемая после синтаксического анализа текста оператора/,
- 5/ магазин операций,
- 6/ магазин операндов,
- 7/ магазин параметров,
- 8/ магазин операторов цикла,
- 9/ таблица ошибок,
- 10/ обменный буфер ввода /для текста на языке фортран/,
- 11/ обменный буфер вывода /для текста на языке автокод, получаемого в результате трансляции/.

В рассматриваемом варианте транслятора поля 1-3, 9, 10 расположены в первом кубе М03У, остальные - в нулевом кубе М03У.

Таблица величин, куда заносятся переменные или константы, встречающиеся в тексте на языке фортран, представляет собой совокупность массивов переменной длины, расположенных последовательно. Каждой величине /переменной или константе/ соответствует один массив. Первое машинное слово массива содержит адрес следующего массива в таблице, а также ряд флагов /признаков/, определяющих структуру остальной части массива.

Таблица макрокоманд представляет собой последовательность двух- и четырехадресных макрокоманд. Адреса макрокоманд суть ссылки на массивы табличных величин. Макрокоманды таблицы являются исходной информацией для макроассемблера при генерации исходного текста транслируемой программы.

Таблица оператора представляет собой результат предварительной обработки фрагментов оператора языка фортран, выполняемой в процессе лексического анализа оператора. Задача лексического анализа - разбиение оператора на ряд фрагментов. Каждый фрагмент является текстовым файлом /последовательностью символов/ и представляет собой основной символ или величину. Основным символом является знак вида

+ - / ( ) = ,

а также конструкции для логических операций, например

.OR. .AND. .NOT. и т.д.

Под величиной в данной работе понимается переменная /идентификатор/ или константа языка фортран. Каждому машинному слову таблицы оператора соответствует основной символ или величина, получаемые в результате лексического анализа. В данном варианте транслятора таблица оператора составляется для арифметических выражений и списка оператора ввода-вывода языка фортран. Величина в таблице оператора есть адрес в таблице величин.

Более подробное описание таблиц транслятора дано в разделе 4.

## 2. КРАТКОЕ ОПИСАНИЕ ОСНОВНЫХ АЛГОРИТМОВ ТРАНСЛЯЦИИ

Процесс трансляции производится в два просмотра транслируемой информации. При первом просмотре подпрограмма на языке фортран преобразуется в таблицу макрокоманд и таблицу величин. Генерация автокодного текста выполняется путем просмотра полученных таблиц после обработки оператора END блоком "END-процессор" /остальные блоки транслятора работают при первом просмотре/.

Трансляция при первом просмотре производится в соответствии со следующей схемой:

- чтение оператора,
- опознавание оператора,
- лексический анализ,
- синтаксический анализ,
- генерация макрокоманд.

При чтении оператора выделяется его метка, и текст самого оператора получается в поле оператора в коде ГОСТ. При этом учитывается, что оператор может находиться на нескольких текстовых картах или занимать часть карты /если операторов на карте несколько/. Есть возможность чтения образов карт текста на языке фортран, пробитых в коде ГОСТ или в телетайпном коде.

Опознавание оператора выполняется после чтения его первого символа /отличного от пробела/. При этом производится просмотр опознающего списка операторов языка фортран, которые могут начинаться с этого символа. Каждый элемент списка содержит опознающий фрагмент оператора языка фортран в коде ГОСТ, адреса входа и выхода для подпрограммы, осуществляющей трансляцию этого оператора, ссылку на следующий элемент списка и дополнительную информацию. Если, например, оператор начинается с символа "D", то в соответствующем списке содержится 4 элемента для операторов "DO", "DATA", "DIMENSION" и "DOUBLE PRECISION".

Если опознающий список для символа пуст /например, оператор начинается с символа "Z"/ или не произошло опознавания при просмотре данного списка, то оператор рассматривается как арифметический.

В случае, если опознавание оператора по начальному фрагменту произошло, но при дальнейшем анализе по соответствующей подпрограмме были обнаружены синтаксические ошибки, оператор считается неопознанным, и продолжается его дальнейшее опознавание с помощью опознающего списка.

Переключатель производит передачу управления на подпрограмму обработки оператора языка фортран, опознавание которого произошло с помощью опознающего списка. Каждая программа обработки вначале делает синтаксический анализ оператора языка, затем /если в процессе синтаксического анализа не было обнаружено ошибки/ выполняется трансляция с оператора в серию макрокоманд.

Блоки обработки арифметических операторов отличаются от блоков обработки неарифметических операторов тем, что в процессе синтаксического анализа арифметических операторов /содержащих арифметические выражения/ формируется таблица операторов, транслируемая затем в серию макрокоманд, тогда как при обработке неарифметических операторов формирование таблицы не производится.

Арифметический блок преобразует фрагменты таблицы оператора, соответствующие арифметическим выражениям, в серию макрокоманд. При этом используется стандартный алгоритм преобразования арифметического выражения в инверсную польскую запись с последующей интерпретацией элементов польской записи /см., например, <sup>14/</sup>/.

Отличие рассматриваемого варианта транслятора от транслятора версии 1971 года <sup>12/</sup> состоит, в частности, в том, что вся арифметика вычисления результата индексной функции транслируется не через подпрограмму вычисления индексных выражений, а непосредственно. Кроме того, постоянную часть индексного выражения транслятор вычисляет и выдает в виде постоянной добавки к адресу операнда. Все это повышает скорость работы транслированной программы и сокращает место в оперативной памяти, занимаемое программой. Более подробно о работе арифметического блока см. в разделе 3.

Лексический анализ оператора языка фортран выполняется блоками, обращение к которым производится из подпрограмм синтаксического анализа оператора. При этом текст оператора языка фортран рассматривается как текстовый файл. В процессе лексического анализа производится разбиение его на текстовые файлы-фрагменты, каждый из которых соответствует идентификатору / в смысле языка алгол-60/, константе языка фортран или основному символу /знаку или логической операции между точками/.

Чтение символа из текстового файла определяется двумя указателями - номером позиции текущего символа и номером позиции символа, с которого начинается файл-фрагмент. Процесс опознавания синтаксической единицы языка начинается с обращения к подпрограмме открытия файла. При этом указатель текущего символа принимает значение указателя начала файла-фрагмента, и чтение символов начинается с начала файла-фрагмента. Если в процессе опознавания произошла ошибка /фактический синтаксис файла-фрагмента не соответствует синтаксису, который пытается опознать программа синтаксического анализа/, то производится выход из подпрограммы синтаксического анализа. Если ошибки не произошло, то в конце подпрограммы синтаксического анализа происходит обращение к подпрограмме лексического анализа, выполняющей закрытие файла-фрагмента. При этом указатель начала файла-фрагмента принимает значение указателя текущего символа, т.е. начала файла-фрагмента, следующего за опознанным.

Описанная конструкция программ синтаксического анализа позволяет в случае ошибки при опознавании конкретной конструкции языка автоматически возвращаться к чтению того же самого файла-фрагмента при переходе к программе синтаксического анализа альтернативной конструкции языка.

Блок лексического анализа включает в себя следующие подпрограммы:

- 1/ подпрограмму открытия файла-фрагмента,
- 2/ подпрограмму закрытия файла-фрагмента перед прочитанным символом,
- 3/ подпрограмму закрытия файла-фрагмента после прочитанного символа,
- 4/ подпрограмму опознавания констант языка фортран,
- 5/ подпрограмму опознавания переменных /идентификаторов/ языка фортран,
- 6/ подпрограмму чтения знаков и основных символов языка фортран.

Подпрограмма открытия файла-фрагмента имеет два режима работы - режим установки указателя файла по номеру символа в операторе и режим установки указателя по предыдущему закрытию файла. Поскольку программа во втором режиме работает значительно чаще, чем в первом, она выполняется за минимальное количество машинных команд, что приводит к сокращению времени, затрачиваемому на синтаксический и лексический анализы операторов языка фортран.



### 3. ОБРАБОТКА АРИФМЕТИЧЕСКИХ ВЫРАЖЕНИЙ В ТРАНСЛЯТОРЕ

Обработка арифметических выражений осуществляется арифметическим блоком в два этапа. Вначале производится синтаксический анализ арифметического выражения и формируется часть таблицы оператора, соответствующая анализируемому арифметическому выражению. При этом выполняются следующие преобразования фрагментов арифметических выражений:

- "=" → "("
- "," → ")" / ";" - символ конца оператора/
- "\*\*" " ^ " / возведение в степень/
- "F(" "F ( <sub>2</sub> " / ° - функциональная композиция/
- "(+ " "(0+"
- "(- " "(0-"
- ".NOT." "O.NOT."
- )(" ") \* ("

В данном случае символ " ^ " означает операцию возведения в степень, "\*" - операцию умножения, "<sub>2</sub>" - первый параметр, "°" - операцию функциональной композиции /взятие функции от параметров/, "→" - символ операции соответствия.

На втором этапе производится просмотр и преобразование строк таблицы оператора в польскую инверсную запись с одновременной интерпретацией элементов этой польской записи. Преобразование в польскую запись производится с помощью магазина операций в соответствии со следующими правилами:

1. Если на входе имеется строка таблицы оператора, соответствующая величине /переменной или константе/, то она отправляется на выход /на интерпретатор польской записи/.
2. Если на входе - строка, соответствующая символу - открывающей скобке, то она отправляется в магазин операций.
3. Если на входе - строка для закрывающей скобки, то из магазина на выход извлекаются накопленные там ранее символы операций по первую открывающую скобку включительно, после чего строка для закрывающей скобки отправляется на выход.
4. Если на входе имеется строка таблицы оператора, соответствующая операции, то анализируется приоритет операции, находящейся в магазине с операцией на входе. Из магазина на выход выталкиваются все операции, приоритет которых не меньше приоритета операции на входе /множество таких операций может быть пустым/. После выталкивания операций из магазина операция на входе засылается в магазин операций. Приоритеты операций строго возрастают для последовательности основных символов:

, .OR. .AND. .NOT. .EQ. + \* ^ °

В данном случае "°" означает операцию "параметр". Приоритеты для следующих групп основных символов совпадают:

.GT. .GE. .LT. .LE. .NE. .EQ.

+ -

\* /

^ ,

Элементы каждой группы расположены в одной строке.

Интерпретация элементов польской записи производится следующим образом:

1. Если на вход интерпретатора поступает величина, то она поступает в магазин операндов.

2. Если на входе имеется операция "Параметр", то берется величина из магазина операндов и пересылается вместе с признаком операции /"Параметр" или "Первый параметр"/ в магазин параметров. Если в магазине операндов находится служебная рабочая ячейка вида  $A * I$ , где  $I$  - целое число, то производится генерация команды вида

$A * I \rightarrow B * J$ ,

где  $B * J$  - служебная рабочая ячейка, в которой хранится результат вычисления параметра. После этого в магазин параметров пересылается адрес в таблице величин для ячейки  $R * I$ , или простой параметр, если он - простая переменная, вместе с признаком параметрической операции. Целые числа  $I$  и  $J$  указывают на количество рабочих ячеек, находящихся в соответствующих магазинах.

3. Если на входе имеется операция функциональной композиции "°", то из магазина операндов извлекается имя функции, генерируется макрокоманда  $F( ) \rightarrow A * I$  вычисления функции. Затем из магазина параметров в служебный магазин пересылаются параметры по ближайший "первый параметр" включительно. После этого служебный магазин исчерпывается, и генерируются макрокоманды для параметров функции, начиная с первого параметра.

4. Если на входе имеются операции, соответствующие открывающей и закрывающей скобкам, то эти операции игнорируются.

5. Для остальных операций из магазина операндов последовательно извлекаются второй и первый операнды, затем выдаются макрокоманды вида

$V1 .OP. V2 \rightarrow A * I$ ;

где  $V1$  и  $V2$  - первый и второй операнды, .OP. - код операции из польской записи,  $A * I$  - ячейка, где хранится результат выполнения операции. Этот результат засылается в магазин операндов.

Формирование программы вычисления индексной функции выполняется во время интерпретации операции "°" /см. пункт 3/. При этом результат вычисления функции хранится в виде ссылки в таблицу величин на составную величину вида F,I, где F - имя функции /базовый адрес/, I - результат вычисления индексного выражения.

При наличии операндов разных типов, а также операндов - формальных параметров или переменных с индексами /результатов вычисления индексных выражений/, вместо макрокоманды

```
Vi .OP. V2 V3;
```

генерируются макрокоманды

```
V1 → R * 1;
```

```
V2 → R * 2;
```

```
V1 .OP. V2 R * 3;
```

```
R * 3 → V3;
```

При этом преобразование типов величин производится следующим способом:

Рабочие ячейки  $R * I$  / $I = 1, 2, 3$ / имеют один и тот же тип /максимальный из типов операндов  $V1$  и  $V2$ /. Преобразование величин из одних типов в другие выполняют макрокоманды  $V1 \rightarrow R * I$ ; / $I = 1, 2$ / и  $R * 3 \rightarrow V3$ ;

Типы в данном варианте транслятора имеют следующие числовые значения:

тип "Булевский" имеет значение "0",

тип "Целый" имеет значение "1",

тип "Вещественный" имеет значение "2",

тип "Двойная точность" имеет значение "3",

тип "Комплексный" имеет значение "4".

Рабочие ячейки вида  $A * I$  могут иметь переменный тип.

В процессе трансляции производится преобразование макрокоманд вида

```
V1 .OP. V2 → A * I;
```

```
A * I → V3
```

в макрокоманду вида

```
V1 .OP. V2 → V3;
```

#### 4. СТРУКТУРА ТАБЛИЦ В ТРАНСЛЯТОРЕ

Таблица величин формируется транслятором в процессе синтаксического анализа оператора. Каждой величине, т.е. переменной или константе, встречающейся в теле оператора языка фортран,

соответствует массив переменной длины в таблице величин. После занесения величины в таблицу она характеризуется адресом первого машинного слова в массиве таблицы величин.

Величины в таблице могут быть переменными, константами и шкалами /составными величинами/.

Первое машинное слово для массива величины /переменной или константы/ содержит адрес - ссылку на начало следующего массива в таблице, а также тип этой величины.

Первое машинное слово для переменной содержит следующие флаги /признаки/:

1/ признак массива постоянной длины,

2/ признак формального параметра,

3/ признак величины DIMENSION,

4/ признак величины COMMON.

Кроме того, в первом машинном слове для переменной содержится информация о количестве машинных слов, занимаемых этой переменной, если она простая, или число машинных слов, занимаемых элементом массива, если переменная - массив. Идентификатор массива /не больше шести символов/ занимает второе машинное слово массива величины. Если переменная - массив постоянной длины, то третья ячейка содержит фактическое количество машинных слов, отводимое под эту величину, а также ссылку на таблицу для идентификатора "B" и восьмеричной константы "C", если для рассматриваемой величины справедливо соотношение

$$A : EQU, B + C$$

/в терминах автокода компилирующей системы БЭСМ-4<sup>1/1</sup>/. Если указанная эквивалентность не имеет места, то адреса ссылок для "B" и "C" суть нули. Оставшаяся часть массива переменной отводится под верхние границы его индексов /по три на машинное слово/, если переменная есть массив.

Первое машинное слово для константы содержит признак текстовой константы.

Следующее машинное слово содержит служебную метку - идентификатор константы. Оставшиеся машинные слова содержат саму константу.

Первое /и единственное/ машинное слово для шкалы содержит признак шкалы, два признака восьмеричных адресов и два 12-разрядных адреса, составляющих рассматриваемую шкалу. Как правило, шкала  $A, I$  представляет величину  $A(I+1)$ . Наличие признака восьмеричного адреса означает, что соответствующая часть шкалы есть восьмеричный адрес, в противном случае адрес есть ссылка на величину в таблице величин.

Таблица макрокоманд содержит двух- и четырехадресные макрокоманды. Код макрокоманды представляет собой четырехразрядное машинное число. Адреса макрокоманд суть ссылки на таблицу величин. Каждая строка /машинное слово/ макрокоманды содержит два адреса и код макрокоманды. В четырехадресной макрокоманде ма-

шинное слово для двух последних адресов содержит нулевой код.

Таблица оператора состоит из строк, в каждой из которых содержится код операции или код величины, приоритет операции, ссылка на таблицу величин /для строки с кодом величины/ и номер позиции для величины или операции в тексте оператора языка фортран /для диагностики/.

## 5. ДИАГНОСТИКА

В процессе синтаксического анализа оператора языка фортран место ошибки определяется номером символа в тексте этого оператора. Номера позиций, где встретились синтаксические ошибки, накапливаются в таблице ошибок. Если в процессе опознавания оператора возможна альтернатива, то таблица ошибок очищается, и опознавание продолжается в соответствии с новой альтернативой. Например, такая ситуация возникает в случае опознавания оператора DO I I=1.

Если после обработки оператора были ошибки, то печатается следующая диагностика - название имени ошибочной подпрограммы, относительный номер ошибочного оператора /относительно начала подпрограммы/, начиная с единицы, текст самого оператора и строку указателей ошибочных мест в операторе.

## 6. ЗАКЛЮЧЕНИЕ

Рассматриваемый транслятор был реализован в составе компилирующей системы БЭСМ-4<sup>1/</sup> в 1975 г. и эксплуатировался в ряде организаций.

Скорость трансляции составляет примерно один оператор в секунду, общая длина блоков транслятора составляет около 10000 машинных слов.

## ЛИТЕРАТУРА

1. Загинайко В.А. ОИЯИ, 11-5923, Дубна, 1971.
2. Загинайко В.А. ОИЯИ, 11-5979, Дубна, 1971.
3. Загинайко В.А. ОИЯИ, 11-6005, Дубна, 1971.
4. Болье Л. В сб.: "Языки программирования", "Мир", М., 1972.
5. Мазный Г.Л. ОИЯИ, 11-5974, Дубна, 1971.
6. Загинайко В.А., Силин И.Н. ОИЯИ, Б-11-5914, Дубна, 1966.

Рукопись поступила в издательский отдел  
13 июня 1983 года.

## НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

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

Д3-11787	Труды III Международной школы по нейтронной физике. Алушта, 1978.	3 р. 00 к.
Д13-11807	Труды III Международного совещания по пропорциональным и дрейфовым камерам. Дубна, 1978.	6 р. 00 к.
Д1,2-12036	Труды VI Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1978 /2 тома/	7 р. 40 к.
Д1,2-12450	Труды V Международного семинара по проблемам физики высоких энергий. Дубна, 1978	5 р. 00 к.
Д11-80-13	Труды VII Всесоюзного совещания по ускорителям заряженных частиц, Дубна, 1980 /2 тома/	8 р. 00 к.
Д4-80-271	Труды рабочего совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике, Дубна, 1979	3 р. 50 к.
Д4-80-385	Труды Международной конференции по проблемам нескольких тел в ядерной физике. Дубна, 1979.	3 р. 00 к.
Д2-81-543	Труды VI Международного совещания по проблемам квантовой теории поля. Алушта, 1981	5 р. 00 к.
Д10,11-81-622	Труды Международного совещания по проблемам математического моделирования в ядерно-физических исследованиях. Дубна, 1980	2 р. 50 к.
Д1,2-81-728	Труды VI Международного семинара по проблемам физики высоких энергий. Дубна, 1981.	2 р. 50 к.
Д17-81-758	Труды II Международного симпозиума по избранным проблемам статистической механики. Дубна, 1981.	3 р. 60 к.
Д1,2-82-27	Труды Международного симпозиума по поляризационным явлениям в физике высоких энергий. Дубна, 1981.	5 р. 40 к.
Р18-82-117	Труды IV совещания по использованию новых ядерно-физических методов для решения научно-технических и народнохозяйственных задач. Дубна, 1981.	3 р. 20 к.
Д2-82-568	Труды совещания по исследованиям в области релятивистской ядерной физики. Дубна, 1982.	3 р. 80 к.
Д9-82-664	Труды совещания по коллективным методам ускорения. Дубна, 1982.	1 р. 75 к.
Д3,4-82-704	Труды IV Международной школы по нейтронной физике. Дубна, 1982.	3 р. 30 к.
		5 р. 00 к.

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

Загинайко В.А.

11-83-399

Алгоритмы и структуры данных транслятора с языка фортран для ЭВМ с малым объемом оперативной памяти

Дается описание алгоритмов и структур данных транслятора с языка, близкого к языку фортран-4. Транслятор реализован на ЭВМ с малым объемом оперативной памяти /БЭСМ-4/.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1983

Zaginajko V.A.

11-83-399

The algorithms and Data Structures of FORTRAN Compiler for Computers with Small Main Storage

The algorithms and data structures of FORTRAN compiler are described. The source language is similar to FORTRAN-IV. The compiler has been realized for a computer with small main storage /BESM-4/.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1983

Перевод О.С.Виноградовой.



**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

Загинайко В.А.

11-83-399

Алгоритмы и структуры данных транслятора с языка фортран для ЭВМ с малым объемом оперативной памяти

Дается описание алгоритмов и структур данных транслятора с языка, близкого к языку фортран-4. Транслятор реализован на ЭВМ с малым объемом оперативной памяти /БЭСМ-4/.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1983

Zaginajko V.A.

11-83-399

The algorithms and Data Structures of FORTRAN Compiler for Computers with Small Main Storage

The algorithms and data structures of FORTRAN compiler are described. The source language is similar to FORTRAN-IV. The compiler has been realized for a computer with small main storage /BESM-4/.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1983

Перевод О.С.Виноградовой.