

СООБЩЕНИЯ  
ОБЪЕДИНЕННОГО  
ИНСТИТУТА  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ  
ДУБНА

Д 458

P10-88-458 e

Н.Д.Дикусар, М.Шпытко

АНАЛИЗАТОР ФОРТРАННОГО ТЕКСТА  
ПРОГРАММНЫХ СИСТЕМ

1988

## ВВЕДЕНИЕ

Программные системы (ПС), предназначенные для обработки данных в научных исследованиях, состоят, как правило, из большого числа (>100) программных модулей (ПМ)\*. При развитии, сопровождении или переносе на другие ЭВМ этих систем необходима информация о структуре ПС, логике взаимодействия ПМ, использовании общих (COMMON) блоков данных и т.п. Требуется знать местонахождение ПМ в ПС, взаимные ссылки в тексте между модулями и их зависимости через COMMON-блоки (статика ПС), а также проследить процесс "выполнения" (псевдовыполнения) ПС на уровне вызовов ПМ с учетом использования данных в COMMON-блоках (динамика ПС). Обычно эта информация не документируется. Некоторая ее часть могла бы быть полученной при трансляции и загрузке ПС, однако в полном объеме ее можно получить только по фортранному тексту листинга.

На практике трансляция и загрузка не всегда выполнимы, а "выуживание" подобной информации по листингу больших программ становится весьма трудоемким занятием даже для самих разработчиков этих систем. Особенно это относится к программам, написанным не в соответствии с правилами хорошего программирования (spaghetti code).

Ряд авторов<sup>1,2/</sup> подчеркивают сложность проблем, связанных с сопровождением математического обеспечения, и предлагают методику и различные программные инструменты для решения этих задач.

Проблема анализа фортранных программ рассматривается в различных подходах. Частным подходом является анализ отдельных характеристик программы, связанных с проверкой обращений к подпрограммам или функциям<sup>3/</sup> и т.п. Другой подход<sup>4/</sup> включает в себя более сложный анализ (в том числе динамику) программ путем внедрения дополнительных инструкций в код программы и ее реального выполнения.

Большой интерес представляет возможность исследования динамики и структуры программы непосредственно по ее тексту, без выполнения самой программы.

---

\*Понятие "модуль" (program unit) — относительно независимая часть программы. Относится к SUBROUTINE, FUNCTION, BLOCK DATA, стандартным функциям и внешним процедурам.

В настоящей работе сделана попытка программного анализа фор-транного текста ПС с целью изучения ее структуры и логики взаимодей-ствия модулей. Для этого были разработаны и созданы программы AFORT (analyzer of fortran text), которые обеспечивают:

- 1) синтаксическое редактирование и структурную распечатку текста программы (pretty code);
- 2) анализ модульной структуры ПС, построение статике и динами-ки использования ПМ;
- 3) дополнительную информацию, необходимую при преобразовании программы на фортране в программы на другие (PASCAL-подобные) структурные языки;
- 4) помощь в подготовке документации отлаженных программных систем.

Результаты анализа подготавливаются в соответствии с указанной при обращении строкой запроса на основе "просмотра" исходного фор-транного текста ПС и выводятся на печать в удобном для работы формате.

## 1. СТРУКТУРА АНАЛИЗАТОРА

Программный анализатор (AFORT) состоит из двух функциональ-но-независимых программ (рис.1): синтаксически ориентированного редактора (syntax analyzer-SAN) и собственно анализатора (FAN).

### 1.1 Редактор

Программа SAN (по своему назначению близка к <sup>15/</sup>) выполняет функцию спецредактора для создания файлов и листингов фортранного текста ПС в удобном для чтения формате (pretty print).

Алгоритм программы выполняет:

- а) форматирование входного текста в соответствии с уровнем опи-сания циклов (операторов "DO"),
- б) выделение некоторых элементов текста через "дважды пропеча-танный" шрифт. Это относится к строкам начала и конца диапазона циклов, именам модулей (как описание, так и обращение к ним), а так-же названиям управляющих операторов фортрана типа "GO TO" и т.п.,
- в) преобразование строк текста ПС с учетом разделителя операто-ров (# или \*),
- г) замену операторов PRINT/PUNCH операторами WRITE(6...)/WRI-TE(7...),
- д) отключение выдачи листинга выходного файла,
- е) выдачу оглавления листинга с именами модулей ПС.

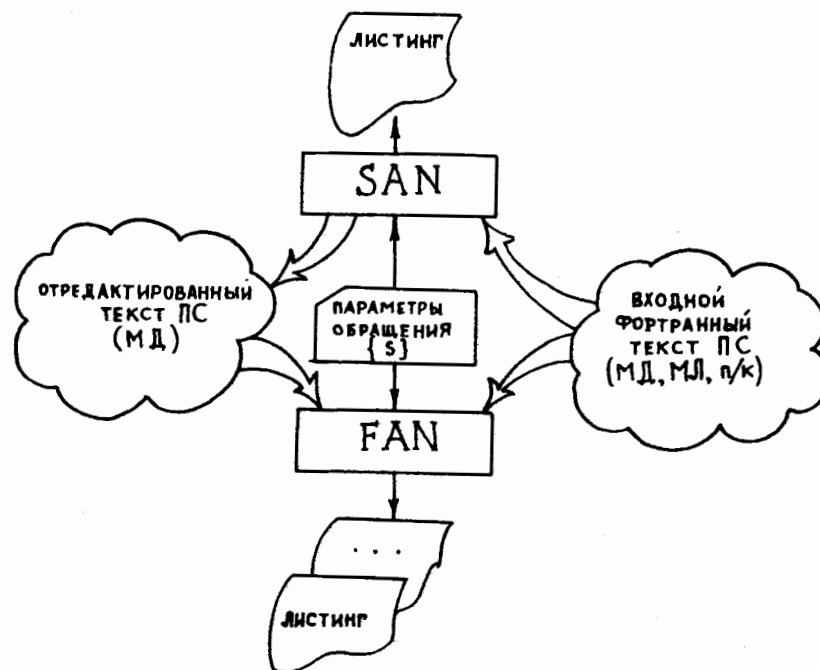


Рис.1. Схема использования программ AFORT.

### 1.2. Анализатор

Выделяются три функциональные группы анализа фортранного текста, позволяющие получить информацию о статике, динамике и струк-туре программной системы.

Большинство результатов анализа текста, которые выдаются на пе-чать в виде списков, таблиц, диаграмм, графов и т.п., не требуют спе-циальных комментариев (рис.2). Поэтому ниже приведены подробные характеристики только для отдельных результатов.

#### 1.2.1 Статика

В режиме просмотра исходного программного текста ПС анализатор распознает описание модулей и переменных, а также находит соотноше-ние между ними в соответствии с обращениями к модулям и использо-ванием COMMON-блоков. Этот этап завершается построением статичес-кого "дерева" ПМ анализируемой ПС в виде

S P E C I F I C A T I O N: 8

CROSS-REFERENCE TABLE FOR COMMON BLOCKS:

COMMON/M O D U L E S:

M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1
0	1	2	3	4	5	6	7	8									

#	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
CC	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
C1	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
C2	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
C3	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
C4	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
C5	<	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
C6	+	+	+	+	!	!	!	!	!	!	!	!	!	!	!	!	!
C7	+	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
C8	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
C9	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
C15	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
C14	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!

# -UNNAMED COMMON BLOCK  
 ! -STATIC DECLARATION OF MEMORY  
 < -REDUNDANT

S P E C I F I C A T I O N: 9

INTERDEPENDENCE OF MODULES IN RESPECT OF COMMON BLOCKS

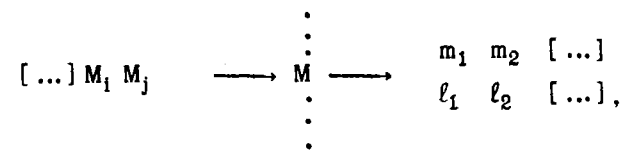
MODULE/M O D U L E S:

M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1

M0	-	-	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X
M1	X	-	-	X	-	X	-	X	-	X	-	X	-	X	-	X	-
M2	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X
M3	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X
M4	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X
M5	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X
M6	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
M7	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X
M8	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
M9	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X	-	X
M10	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
M11	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
M12	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!

X -DIRECT INTERDEPENDENCE

Рис.2. Статика программной системы: а) использование COMMON-блоков, б) взаимозависимость модулей через COMMON-блоки.



где M — имя модуля, M<sub>1</sub>, M<sub>j</sub> — имена модулей, в которых было обращение к M, m<sub>1</sub>, m<sub>2</sub>,... — имена вызываемых в M модулей, l<sub>1</sub>, l<sub>2</sub>,... — номера строк в тексте, где происходят эти вызовы.

На следующем шаге воспроизводится динамика ПС, отражающая уровни обращения к ПМ и использование общих блоков данных.

1.2.2. Динамика

Динамика ПС представлена в трех видах (11, 12 и 13 спецификации). Например, информация для 11 спецификации печатается в виде

s: level : ... m<sub>1</sub>/L<sub>1</sub>/ I<sub>1</sub> I<sub>2</sub> [:I<sub>3</sub>],

где s— шаг "выполнения" (в смысле обращения к ПМ), level — текущий уровень "выполнения", m<sub>1</sub>— имя вызываемого ПМ, l<sub>1</sub> — номер строки в тексте, содержащей обращение к ПМ, I<sub>1</sub> = \* — идентификатор обращения к ПМ в цикле (циклах), I<sub>2</sub> =(F/I/0) — идентификатор операторов доступа к файлам ввода/вывода, I<sub>3</sub>= comment — описание функции ПМ.

Каждый шаг обращения к ПМ содержит (на адекватном уровне) имя вызываемого ПМ и номер строки текста, где это обращение происходит (рис.3). Последний элемент протокола этой спецификации необязателен. Так, описание функции модуля появится только при введении комментария в сегментах SUBROUTINE, FUNCTION и т.п. в виде

C: comment,

где comment — описание функции модуля.

Кроме того, протокол указывает на случаи, когда данный вызов ПМ происходит в теле цикла (циклов), а также на то, что в данном ПМ встречаются операторы ввода/вывода или доступа к файлам (READ, WRITE, OPEN, CLOSE и т.п.).

Страница продолжения листинга начинается с заголовка, содержащего имена модулей, "выполненных" на соответствующих уровнях. Это удобно при работе с листингом.

На рис.4 представлен граф передач управления между ПМ, построенный анализатором при задании спецификации 12.

Информация об использовании COMMON-блоков в процессе "выполнения" исследуемой ПС (спецификация 14) представлена в виде

SPECIFICATION: 11

"DYNAMIC S" /VERSION 1/

NOTICE: "STEP" MEANS PERFORMANCE OF CALL

STEP:[ XX ]: L E V E L: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```

1:[ 1 ]: MO (I/O)
2:[ 2 ]: . M1 / 12/ :M1-FUNCTION DESCRIPTOR
3:[ 3 ]: . . M2 / 44/ :M2-FUNCTION DESCRIPTOR
4:[ 4 ]: . . . M4 / 54/ :M4-FUNCTION DESCRIPTOR
5:[ 4 ]: . . . M5 / 55/ :M5-FUNCTION DESCRIPTOR
6:[ 5 ]: . . . M6 / 87/ (I/O) :M6-FUNCTION
7:[ 6 ]: . . . M12 / 94/
8:[ 5 ]: . . . M10 / 88/
9:[ 6 ]: . . . M15 / 146/ ≡
10:[ 6 ]: . . . M4 / 148/ :M4-FUNCTION
11:[ 6 ]: . . . M17 / 149/
12:[ 5 ]: . . . M18 / 89/
13:[ 4 ]: . . . M17 / 56/
14:[ 3 ]: . . . M4 / 45/ ≡ :M4-FUNCTION DESCRIPT
15:[ 3 ]: . . . M7 / 46/ (0) :M7-FUNCTION DESCRIPT
16:[ 4 ]: . . . M11 / 107/ ≡
17:[ 5 ]: . . . M5 / 155/ :M5-FUNCTION DESCRIPT
  
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  
 . . . . . . . . . . . . . . . .

M M M M  
 0 1 7 1

```

61:[ 4 ]: . . . M18 / 89/
62:[ 3 ]: . . . M14 / 121/
63:[ 3 ]: . . . M15 / 122/
64:[ 2 ]: . . M14 / 22/ ≡
65:[ 2 ]: . . M15 / 23/ ≡
66:[ 2 ]: . . M16 / 25/
  
```

WHERE: ≡ -EXECUTED IN RANGE OF LOOP/S/  
 ; -MODULE-FUNCTION DESCRIPTOR,  
 /LINE/ -LINE-NUMBER OF CALL,  
 (F/I/O) -DATA-ACCESS IDENT.:  
 F -FILE PREPARATION /INQUIRE, OPEN, CLOSE,  
 I -DATA INPUT /READ/  
 O -DATA OUTPUT /WRITE, PRINT, PUNCH/;

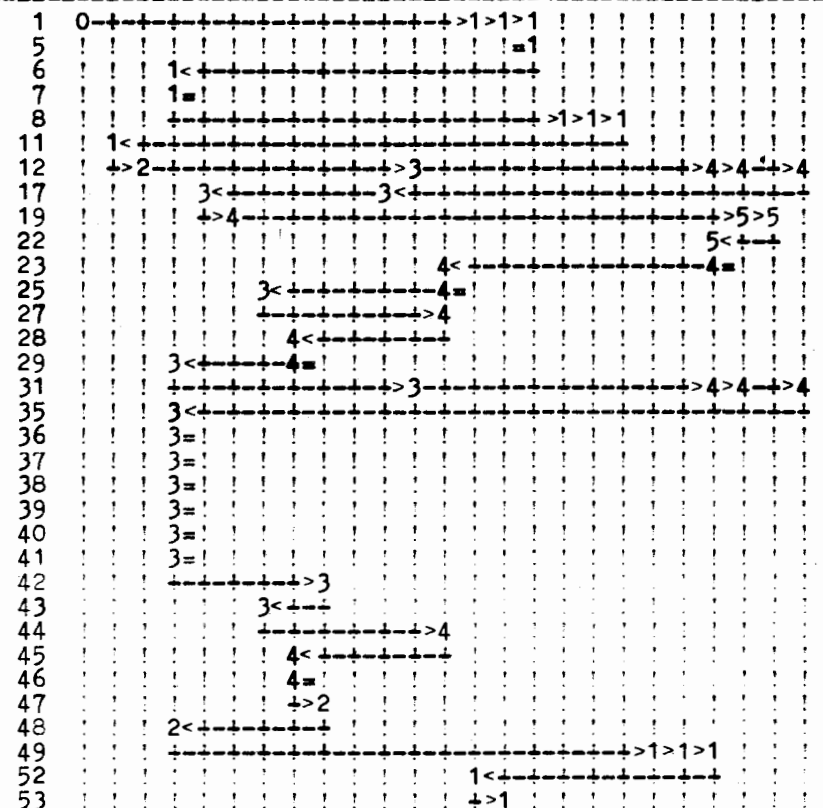
Рис.3. Динамика программной системы (фрагмент выдачи по спецификации 11.

SPECIFICATION: 12

"DYNAMIC S" /VERSION 2/

STEP: M O D U L E S:

S H T N I G O B O L C N D F F U F O G P C C C G M  
 W O P O N E U C U O O E E T D Z I P E U L L L T Y  
 A L T T D T P T D T K P W C I A E L E T T O O N R  
 P D M I A T S D C F S M O M T R E N S F E  
 E C C T T I O R E T D E E O W M E E C  
 R 2 E A R S P M T A M T



WHERE: > X < -LEVEL OF EXECUTION: 0, 1, 2, ...  
 = ' -THIS SAME MODULE  
 ←-----→ -TRANSFER OF CONTROL  
 ? -RECURRENCE

Рис.4. Динамика ПС (передача управлений) — фрагмент выдачи по спецификации 12 для ПС SWAP из [4].

протокола с указанием последовательности шагов "выполнения" программы. Описание для этого протокола подготавливается спецификацией 13 (рис.5, имена вызывающих ПМ "дважды пропечатаны").

Анализатор учитывает положение точек ENTRY в теле подпрограммы, сообщает о наличии рекурсивных обращений, а также вычисляет общий объем оперативной памяти, необходимый для размещения COMMON-блоков (статика и динамика), и выдает статистику использования отдельных типов операторов фортрана.

### 1.2.3 Структура

Преобразование фортранного текста в текст PASCAL-подобного структурного языка связано с некоторыми трудностями. Чтобы определить структуру программы, необходимо сначала определить реальные уровни описания ее модулей и COMMON-блоков.

На первом этапе анализа структуры, опираясь на динамику программы (спецификации 11-14), анализатор определяет существующие уровни описания модулей. Заметим, что после анализа динамики уровни описания уже частично определены. После этого, в соответствии с правилами структурного программирования (structured programming), преобразовываются уровни описания модулей. Каждое перемещение данного описания ПМ на другой (всегда нижний) уровень вызывает коррекцию уровней описания всех модулей, связанных с данным модулем и описанных на высших уровнях. На последнем этапе, согласно полученной структуре программы, выполняется анализ использования COMMON-блоков в каждом модуле.

В результате описанных выше действий получаем вариант PASCAL-подобной структуры исследуемой программной системы, в том числе предполагаемые уровни и места (в каком модуле) описаний COMMON-блоков (рис.6).

Остальные спецификации (начиная с 20) являются расширением основных и указывают на анализ описания переменных, типов, вызовов функций, вычисление длины COMMON-блоков и т.п.

### 1.2.4 Обращение

При обращении к анализатору указывается строка (набор) спецификаций {S}. Спецификации задаются на шкале индексов (номеров), которая на экране терминала или на перфокарте представлена в виде

1 2 3 ... 0 1 2 3 ... 0 1 2 3 4 5.

Чтобы указать желаемый набор {S}, достаточно на месте соответствующих индексов шкалы поставить символ "X". Например, в строке

STEP: [XX]:	1	2	3	4	5	6	STEP: [XX]:
1:	M0	M1	M2	M4	M6	M12	1:
2:	M0	M1	M2	M5	M10	M15	2:
3:	M0	M1	M2	M5	M10	M15	3:
4:	M0	M1	M2	M5	M10	M17	4:
5:	M0	M1	M2	M5	M10	M18	5:
6:	M0	M1	M2	M5	M10	M18	6:
7:	M0	M1	M2	M5	M10	M17	7:
8:	M0	M1	M2	M5	M10	M15	8:
9:	M0	M1	M2	M5	M10	M15	9:
10:	M0	M1	M2	M5	M10	M17	10:
11:	M0	M1	M2	M5	M10	M17	11:
12:	M0	M1	M2	M5	M10	M17	12:
13:	M0	M1	M2	M5	M10	M17	13:
14:	M0	M1	M2	M5	M10	M17	14:
15:	M0	M1	M2	M5	M10	M17	15:
16:	M0	M1	M2	M5	M10	M17	16:
17:	M0	M1	M2	M5	M10	M17	17:

..  
 < -REDUNDANT,  
 ! -NOT USED,  
 X -USED BY SOME MODULES EXECUTED ON THIS STEP,  
 0 -USED BY ALL MODULES EXECUTED ON THIS STEP,  
 \* -USED BY LEVEL-[I-1] AND LEVEL-[I] MODULES,  
 ..

Рис.5. Динамика ПС с использованием COMMON-блоков (фрагмент выдачи по спецификациям 13 и 14).

SPECIFICATION: 17.2

PROPOSED "PASCAL-LIKE" STRUCTURE OF THE PROGRAM:

LEVEL:

	0	1	2	3	4
M0	[M1		[M7	],	
	M11		,		
	M2		,		
	M3		[M9	],	
	M8		,		
	M19		,		
	M4		,		
	M5		[M10	[M12	],
			M13	],	
	M6		]		

SPECIFICATION: 17.4 /SUPPLEMENT TO SPECIFICATION 17.2/

LEVELS OF COMMONS DESCRIPTION:

COMMON: EXISTING: PROPOSED: IN MODULE:

1: 01	:	.	.	.	0	!	.	.	.	0	.	.	M0
2: 02	:	.	.	.	0	!	.	.	.	0	.	.	M0
3: 03	:	.	.	.	0	!	.	.	.	0	.	.	M0
4: 04	:	.	.	.	0	!	.	.	.	0	.	.	M0
5: 05	:	.	.	.	0	!	.	.	.	>+	.	.	M0
6: 06	:	.	.	.	1	!	.	.	.	>0	.	.	M0
7: 07	:	.	.	.	1	!	.	.	.	>0	.	.	M0
8: 08	:	.	.	.	1	!	.	.	.	>0	.	.	M0
9: 09	:	.	.	.	2	!	.	.	.	>1	.	.	M3
10: 010	:	.	.	.	3	!	.	.	.	>2	.	.	M10
11: 011	:	.	.	.	1	!	.	.	.	>0	.	.	M0
12: 014	:	.	.	.	3	!	.	.	.	>1	.	.	M5

NOTE: + -REMAINING,  
> -REMOVED TO THIS LEVEL;

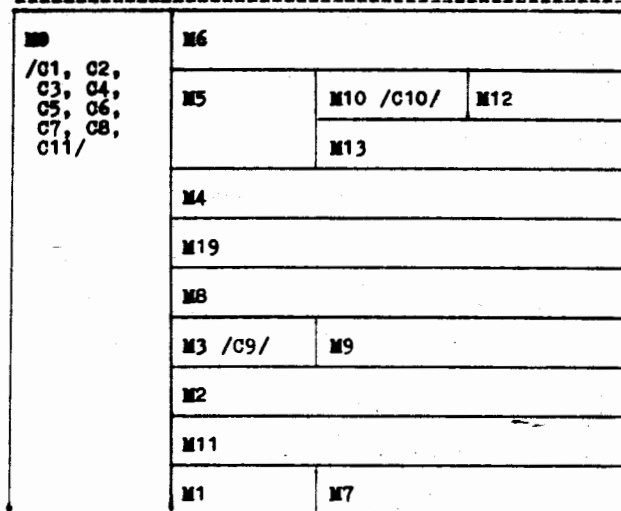


Рис.6. Предлагаемая структура ПС и место описания COMMON-блоков.

X 2 3 ... X 1 2 3 ... X X 2 3 4 X

задан набор спецификаций {S} = {1, 10, 20, 21, 25}.

Полный набор спецификаций для анализатора приведен в табл.1, а на рисунках показаны примеры фрагментов выдачи на листинге.

Таблица 1

Функции AFORT

индекс	спецификация	результаты: содержание и формат
1/S	список модулей ПС (Mi)	Mi mj li
2/S	список внешних имен (mi)	ni mi
3/S	список сегм. FUNCTION (Mi)	ni Mi
4/S	список вх.ENTRY в п/п(Mj)	ni Ei Mj
5/S	список "DO"-циклов (Li)	ni Mj Li li-start li-end
6/S	список COMMON-блоков (Ci)	ni Mj Ci li di
7/S	список перекрестных ссылок ICOMMON-бл. (Ci) на модули (Mj)	Ci Mi1 Mi2 ... [di1 di2 ...]
8/S	таблица перекрестных ссылок ICOMMON-бл. (Ci) на модули (Mj)	Mj (Ij-идентификатор) ... (Mj - MAIN-сегм. ) Ci I1 I2...Ij ... (= - redundant ) ... (=+ - вне MAIN )
9/S	зависимость модулей I программы (Mi, Mj) через ICOMMON-блоки	Mj (Ij-идентификатор) ... (Ij-идентификатор) Mi I1 I2 ...Ij ... (X - зависимость )
10/S	"статическое дерево" I модулей программной сист.	... m_2i m_1i Mi mi1 m2i ... l1i l2i ... m_ji - вызываемый; m_ki - вызываемый ПМ
11/D	динамика программы: I -версия 1 (обращение к ПМ)	si:(Xi): ... mi/li/ I1 I2 I3 где I1=* - вызов mi в пределах цикла(ов) I2=(F/I/O)-в mi есть доступ к данным I3=:comment-дескриптор функции ПМ
12/D	динамика программы: I -версия 2 (граф передачи I управления между ПМ)	... Mi ... Mj ... si: ... X ! ... ! ! ! ... ! ! ! ...
13/D	динамика программы: I -версия 3 (уровни I выполнения ПМ; прилож.к14/D)	si:(Xi): mi1 mi2 ... miXi
14/D	динамика программы: I (использование COMMON-бл.)	Cj ... si:(Xi): I1 I2...Ij ... где Ij= - не используется Ij=X - используется в некоторых ПМ Ij=0 - используется во всех ПМ Ij== - используется в ПМ уровня X-и X

Таблица 1 (продолжение)

15/S	I список использования операторов доступа к данным (I <sub>i</sub> )	I n <sub>i</sub> I <sub>i</sub> L <sub>i</sub> , где I <sub>i</sub> ∈ {INQUIRE, OPEN, CLOSE, READ, WRITE, PRINT, PUNCH, DEFINEFILE, ENDFILE, REWIND, BACKSPACE, WAIT}
16/S	I таблица использования операторов доступа к данным (I <sub>i</sub> ) в ПМ	I I <sub>1</sub> I <sub>2</sub> ... I <sub>12</sub> ... I L <sub>i</sub> M <sub>i</sub> f <sub>i1</sub> f <sub>i2</sub> ... f <sub>i12</sub> , где I <sub>i</sub> - то же что и в 15/S, f <sub>ij</sub> - идентификатор файла для I <sub>i</sub>
17/B	I структура программы: а) данные уровни описания ПМ и COMMON-блоков, б) предлагаемая PASCAL-подобная структура ПС: в) предлагаемые уровни описания COMMON-блоков	I ПМ и COMMON-бл. описаны на данном уровне и их связь с другими уровнями; I запись MOD[M1[M2[M3], ..., M7[M8[M9]]]] I представляет следующую структуру программы: I MO I M1 I M2 I I I M3 I I I I I I I I I M7 I M8 I M9 I I I I I
18/D	I гистограмма использования ПМ: S-статика, D-динамика без циклов (кол-во обращений к ПМ), n-число вызовов ПМ, ns-статика (=), nd-динамика (-)	I 1 2 3 4 5 6 7 8 ... I M <sub>i</sub> (ns/nd) :=====

где:  
S, D, B - идентификатор функциональной группы:  
S - статика, D - динамика, B - структура.  
n - текущий номер;  
M - имя определенного модуля (SUBROUTINE, FUNCTION, BLOCK DATA);  
E - имя входа /ENTRY/ в подпрограмму;  
m - имя вызываемого модуля (M, E или внешняя подпрограмма);  
C - имя COMMON-блока;  
d - длина COMMON-блока;  
L - идентификатор метки оператора;  
l - номер строки текста программы (декларации, вызова модуля и т.п.);  
s - текущий шаг "выполнения программы" (в смысле вызовов к ПМ);  
X - текущий уровень "выполнения программы" (вложение вызовов);

## 2. НЕКОТОРЫЕ ХАРАКТЕРИСТИКИ АНАЛИЗАТОРА

Программы анализатора SAN и FAN ориентированы на работу с текстом различных версий фортрана (FORTRAN-IV, FORTRAN-77, ANSI-FORTRAN). Это относится к типам переменных, ключевым словам и стандартным функциям.

2.1 Язык программирования: программы AFORT написаны на языке PASCAL-8000 с минимальными отклонениями от стандарта.

2.2 Параметры обращения к программам: параметры обращения к редактору (SAN) задаются в виде строки из 5 символов, где i-я позиция указывает на i-ю функцию (i = 1, ..., 5). Параметры обращения к анализатору (FAN) задаются строкой записи из 25 символов, где позиция i-го символа указывает на i-ю спецификацию (см. табл.1). Реализация выбранной функции (для SAN) или спецификация (для FAN) указывается символом "X".

2.3 Необходимые ресурсы указаны в табл.2.

Таблица 2

№	РЕСУРС	SAN	FAN
1.	Оперативная память (байт) из них: код программы стек "куча" (heap)	~ 127K ~ 99K ~ 11K ~ 18K	~ 386K ~ 202K ~ 153K ~ 32K
2.	Внешняя память на МД	входной и выходной файлы данных	входной файл рабочие файлы
3.	Печатающее устройство (символов в строке)	120 55	120 44
4.	Время выполнения* (с)		(из них 16 - анализ текста)

\*центрального процессора EC-1061 для ПС из 40 модулей с фортраным текстом из 4100 строк.

## 2.4. Ограничения (только для FAN)

Максимальное число: описаний модулей	150
всех модулей	200
COMMON-блоков	100
вызовов в одном ПМ (без обращения к стандартным функциям)	60

## 3. ЗАКЛЮЧЕНИЕ

В заключение отметим, что создание программ AFORT было стимулировано работами по развитию математического обеспечения системы ПРОМИС<sup>8/</sup> и ее переносу с ЭВМ CDC-6500 на EC-1061.



## ЛИТЕРАТУРА

1. Dennie Van Tassel, *Program Style, Design, Efficiency, Debugging and Testing*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, U.S.A., 1978.
2. Oliver P., *Software and Benchmarking*. In: *Performance of Computer Installations*. Edited by D. Ferrari. Pub. by North Holland Publishing Company. Proc. of ICPCI78, June 22-23, 1978.
3. Cass T. *Inter Compilation Analyzer for VSFORTRAN 2.2*, CERN Computer Newsletter, No. 191, February-April, 1988.
4. Gordon Lyon, Rona B. Stilman. *A FORTRAN Analyzer*, NBS Technical Note 849, Washington, October 1974.
5. Хасанов А.Н. Методика переноса программ на фортране с ЭВМ CDC-6500 и БЭСМ-6 на ЕС ЭВМ. ОИЯИ 11-83-824, Дубна, 1983.
6. Дикусар Н.Д., Тхай Ле Ханг, Харьюзов М.Р. Фильтрация и сборка событий с МИС на CDC-6500. ОИЯИ P10-12729, Дубна, 1979.

## 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

Рукопись поступила в издательский отдел  
27 июня 1988 года.

НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

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

Д13-84-63	Труды XI Международного симпозиума по ядерной электронике. Братислава, Чехословакия, 1983.	4 р. 50 к.
Д2-84-366	Труды 7 Международного совещания по проблемам квантовой теории поля. Алушта, 1984.	4 р. 30 к.
Д11.2-84-599	Труды VII Международного семинара по проблемам физики высоких энергий. Дубна, 1984.	5 р. 50 к.
Д17-84-850	Труды III Международного симпозиума по избранным проблемам статистической механики. Дубна, 1984. (2 тома)	7 р. 75 к.
Д11-85-791	Труды Международного совещания по аналитическим вычислениям на ЭВМ и их применению в теоретической физике. Дубна, 1985.	4 р. 00 к.
Д13-85-793	Труды XII Международного симпозиума по ядерной электронике. Дубна, 1985.	4 р. 80 к.
Д4-85-851	Труды Международной школы по структуре ядра. Алушта, 1985.	3 р. 75 к.
Д3.4.17-86-747	Труды V Международной школы по нейтронной физике Алушта, 1986.	4 р. 50 к.
—	Труды IX Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1984. (2 тома)	13 р. 50 к.
Д1.2-86-668	Труды VIII Международного семинара по проблемам физики высоких энергий. Дубна, 1986. (2 тома)	7 р. 35 к.
Д9-87-105	Труды X Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1986. (2 тома)	13 р. 45 к.
Д7-87-68	Труды Международной школы-семинара по физике тяжелых ионов. Дубна, 1986.	7 р. 10 к.
Д2-87-123	Труды Совещания "Ренормгруппа - 86". Дубна, 1986.	4 р. 45 к.
Д4-87-692	Труды Международного совещания по теории малочастичных и кварк-адронных систем. Дубна, 1987.	4 р. 30 к.
Д2-87-798	Труды VIII Международного совещания по проблемам квантовой теории поля. Алушта, 1987.	3 р. 55 к.
Д14-87-799	Труды II Международного симпозиума по проблемам взаимодействия мюонов и пионов с веществом. Дубна, 1987.	4 р. 20 к.
Д17-88-95	Труды IV Международного симпозиума по избранным проблемам статистической механики. Дубна, 1987.	5 р. 20 к.

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

Дикусар Н.Д., Шпытко М.  
Анализатор фортранного текста программных систем

P10-88-458

Обсуждается проблема анализа с помощью ЭВМ фортранных текстов программных систем в рамках современных идей практики программирования. Приведено описание программного обеспечения, разработанного и созданного для достижения следующих основных целей: 1) анализ статики и "динамики" программных систем, 2) анализ структуры программы с точки зрения перевода ее фортранного текста на другие PASCAL-подобные языки, 3) подготовка документации отлаженных программных систем.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1988

Перевод авторов

Dikoussar N.D., Shpytko M.  
A Fortran Text Analyzer

P10-88-458

This article deals with the problem of the computer analysis of source FORTRAN programs text, in the light of modern ideas of programming practice. It describes the supporting software, designed and implemented with the following principal aims: 1) analysis of statics and "dynamics" of programs, 2) analysis of the programs structures, with special regard to conversion of FORTRAN text into PASCAL-like structural language and 3) documentation of debugged programs.

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

Communication of the Joint Institute for Nuclear Research. Dubna 1988