



ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

3602/82

2/III-82

5-82-326

А.А.Корнейчук

О ПРОГРАММИРОВАНИИ
НА СТРУКТУРНОМ ДИАЛЕКТЕ ФОРТРАНА

Направлено в Оргкомитет семинара
"Проблемы информатики и ее применения в управлении,
обучении и научных исследованиях"
/София, июнь 1982 г./

1982

1. О ПОЗНАВАЕМОСТИ ПРОГРАММ

Структурный диалект фортрана /СДФ/ ^{1/1/} - язык программирования, представляющий собой фортран, расширенный дополнительными конструкциями в соответствии с идеями структурного программирования ^{2-8/}.

При создании СДФ была поставлена цель: отходя от фортрана как можно меньше, предложить такое его расширение, которое в большей степени, чем это сделано для последних стандартных версий фортрана /типа фортран-77/, учитывало бы современные тенденции в программировании: стремление к уменьшению сложности программ, использование ограниченного числа управляющих структур, нисходящее конструирование программ, самодокументированность программ, и было бы удобным инструментом конструирования познаваемых программ, то есть таких программ, которые могут быть достаточно легко изучены с целью их использования для решения задач на ЭВМ, постановки на ЭВМ и модификации без непосредственных контактов с их авторами.

Познаваемость программы - одно из важнейших ее свойств. "Стало почти невозможно разобраться в том, что делает программа, написанная другим человеком, и практически невозможно отладить до конца такую программу. Возникла ситуация, когда легче заново написать программу, чем разобраться в том, что делает чужая, или немного изменить ее. В результате примерно 80-90% существующих в настоящее время программ функционально дублируют друг друга, а из всего огромного объема накопленного программного обеспечения активно используется только 1-3%" /см. стр.3 в ^{1/1/}.

2. О НЕОБХОДИМОСТИ РАСШИРЕНИЯ ФОРТРАНА

Создание познаваемых программ на фортране, в принципе, возможно и без его расширения. Так, например, одно из средств достижения познаваемости программы - структурное программирование в узком смысле слова, то есть использование ограниченного числа управляющих структур, - может быть обеспечено путем "ручного" моделирования этих структур на фортране ^{4.14/}. Однако путь расширения фортрана соответствующими управляющими структурами с последующим их машинным переводом на фортран представляется более естественным и перспективным.

Уменьшение сложности программы /явное указание логически замкнутых фрагментов программы/ - важнейшее средство достижения познаваемости программы. Традиционно используемое на фортране разбиение программ на подпрограммы /модули/ не совсем отвечает этой цели. Разбиение программы на модули в несколько десятков строк фортрана каждый /именно таким по ряду простых причин - таких, например, как необходимость иметь текст на одной странице - представляется оптимальный размер фрагмента познаваемой программы/ не всегда возможно, не всегда целесообразно и, как правило, связано с необходимостью рутинного повторения описаний общих блоков данных во всех использующих их подпрограммах. Поэтому расширение фортрана простейшими макросредствами /макроопределениями и макровыводами без параметров/ ^{15/} - едва ли не самое необходимое, если стремиться к познаваемости программ.

Нисходящее конструирование программы состоит в том, что программа записывается сначала как один обозримый фрагмент на некотором гипотетическом языке, с которого нет транслятора в коды ЭВМ, и затем постепенно уточняется - до тех пор, пока не окажется записанной на некотором реальном языке /с которого есть транслятор в коды ЭВМ/. Использование упомянутых выше простейших макросредств /текстовых вставок/ позволяет конструировать программу "сверху вниз" как древовидную структуру, постепенно добавляя к ней уточняющие ее фрагменты нижних уровней и почти не меняя при этом фрагменты верхних.

При расширении фортрана более сложными макросредствами обычно имеют в виду другие цели /например, генерацию версий программ ^{16/} /.

3. О САМОДОКУМЕНТИРОВАННОСТИ ПРОГРАММЫ

Будем считать программу самодокументированной, если она содержит в себе исчерпывающую информацию для ее использования /решения с помощью программы задач на ЭВМ/, постановки /подготовки к использованию/ на ЭВМ и модификации /с целью исправления обнаруженных ошибок либо модернизации/. Убедительные аргументы в пользу самодокументированных программ можно найти в ^{17/} /.

Самодокументированность программы - надежное средство обеспечения ее познаваемости: надо лишь позаботиться не только о полноте информации, но и о подходящей форме ее представления.

4. УРОВНИ СДФ

Структурный диалект фортрана, строго говоря, не единственный язык, а семейство языков, называемых уровнями СДФ /СДФ/1, СДФ/2, СДФ/3 ^{1,10/} /.

Уровни СДФ отражают эволюцию СДФ "от фортрана": наиболее близок к фортрану СДФ/1, наиболее удален /в настоящий момент/ СДФ/3.

СДФ/1 представляет собой фортран, расширенный простейшими макросредствами - макроопределениями /СДФ-блоками/ и макровыводами /специальными операторами вызова СДФ-блоков/ без параметров, а также СДФ-комментариями, помещаемыми между СДФ-блоками.

Формальные требования СДФ/1 дополняются неформальным /не контролируемым транслятором "СДФ/1-фортран"/ соглашением о независимости СДФ/1-блоков по управлению /ссылки на метки другого блока, за исключением меток форматов, недопустимы/.

СДФ/2 представляет собой СДФ/1, расширенный обобщенными фортран-операторами, в которых допустимы относительные метки /например, ..1, ..23, ..456/, локализованные в теле блока. Это обеспечивает независимость СДФ/2-блоков по управлению /при условии, что абсолютные метки используются только как метки форматов/.

При программировании на СДФ/2 возможна ступенчатая запись СДФ/2-блоков, отражающая их внутреннюю структуру и поддерживаемая транслятором "СДФ/2 - фортран" в том смысле, что соответствующую ступенчатую запись имеет и результат работы транслятора /фортран-программа/.

СДФ/3 представляет собой СДФ/2, расширенный некоторым не вполне традиционным набором управляющих структур /альтернатива с управляющими условиями, альтернатива с управляющей переменной, цикл с управляющей переменной, цикл без управляющей переменной/ и набором специальных операторов, в числе которых условный и безусловный выходы из блока или цикла, пропуск оставшейся части тела блока или цикла и др.

Фортран-программа будет правильной СДФ/1-программой, если ее заключить в операторные скобки *MAIN ... *END. Правильная СДФ/1-программа является правильной СДФ/2-программой, а правильная СДФ/2-программа - правильной СДФ/3-программой. В этом смысле СДФ/2 включает в себя СДФ/1, а СДФ/3 - СДФ/2.

Промежуточные уровни СДФ /СДФ/1 и СДФ/2/ представляются полезными при описании СДФ, освоении его лицами, владеющими фортраном, и реализации транслятора "СДФ - фортран". Практическое использование СДФ/1 и СДФ/2 в качестве языков программирования возможно, но, вероятно, нецелесообразно.

5. УРОВНИ СДФ И ПОЗНАВАЕМОСТЬ ПРОГРАММЫ

Уровни СДФ /СДФ/1, СДФ/2, СДФ/3/ вполне сопоставимы с рассмотренными выше средствами обеспечения познаваемости программ.

СДФ/1 благодаря наличию СДФ-блоков и спецоператоров их вызова обеспечивает уменьшение сложности и нисходящее конструирование программы без регламентирования ее модульной структуры на Фортране: подпрограмма /модуль/ на Фортране в представлении на СДФ может быть, если это необходимо, древовидной структурой из многих СДФ-блоков.

СДФ/1 благодаря помещаемым между блоками СДФ-комментариям существенно облегчает конструирование самодокументированных программ и согласование с системами машинной подготовки программной документации /см., например, /11/ /.

Относительные метки, введенные в СДФ/2, делают возможным независимое /с точностью до связей по данным/ конструирование СДФ/2-блоков, а также при использовании СДФ/2 в качестве промежуточного языка трансляции независимое преобразование СДФ/3-блоков в СДФ/2-блоки.

Ступенчатая запись, применяемая как локальное /в пределах блока/ средство повышения познаваемости СДФ-программы, оказывается также и глобальным /в пределах Фортран-модуля/ средством повышения познаваемости Фортран-программы.

СДФ/3 - расширение Фортрана, в котором по сравнению с СДФ/1 и СДФ/2 вводится наибольшее число новых изобразительных средств. Непосредственное их назначение - обеспечить познаваемость автономного по управлению, обозримого СДФ-блока.

СДФ/3-программа, как правило, не имеет меток.

Выбор одного из возможных вариантов /выполнение одной из нескольких групп операторов/ задается альтернативой с управляющими условиями вида

```
* IF ... *THEN ... *ELSEIF ... *THEN ... .. *ELSE ... *END
```

/допустимы сокращенные формы типа

```
* IF ... *THEN ... *END; * IF ... *THEN ... *ELSE ... *END
```

и др. / или же альтернативой с управляющей переменной вида

```
* SELECT ... *CASE ... *CASE ... *ELSE ... *END
```

с указанием имен случаев, соответствующих значениям управляющей переменной.

Повторное выполнение группы операторов задается циклом с управляющей переменной вида

```
* LOOP J = 1, 10 ... *END
```

или же циклом без управляющей переменной /бесконечным циклом/ вида

```
* LOOP L ... *END.
```

За исключением альтернативы с управляющими условиями, совпадающей с точностью до ключевых слов с одной из конструкций Фортрана-77 /12, 13/, остальные управляющие структуры нетрадиционны, в частности, благодаря тому, что СДФ/3 допускает структурные переходы /безусловные и условные выходы из цикла и пропуски оставшейся части тела цикла с указанием имени цикла/ вида

```
* IF ... *LEAVE J; *LEAVE L; *IF ... *SKIP L; *SKIP J.
```

В СДФ/3 возможны символические метки и переходы на них, например

```
* LABEL L; * IF (A.LT.B) *GOTO L; *GOTO L,
```

а также спецоператоры обмена с внешними устройствами с неявной ссылкой на формат, задаваемый непосредственно после спецоператора обмена, например

```
* WRITE (NU, *FORMAT) X, Y
```

```
* FORMAT (2F 10.3) .
```

Введенные в СДФ/3 в /1/ спецоператоры перехода на начало блока или цикла вида

```
* ENTER I,
```

как показала практика их использования, ухудшают познаваемость программы и должны быть исключены из СДФ.

Сравнительный анализ СДФ и некоторых известных расширений Фортрана для целей структурного программирования дан в /1/. В числе работ, в которых предлагаются такие расширения, работы /4-6, 18-20/.

6. ОБЩАЯ СТРУКТУРА СДФ-ПРОГРАММЫ

Познаваемая СДФ-программа - текст многоцелевого назначения, используемый не только для получения программы и ее последующего выполнения на ЭВМ, но и как исчерпывающий источник сведений о программе, необходимых для ее использования, постановки на ЭВМ и модификации. Поэтому СДФ-программа должна содержать

- краткое описание /руководство пользователя/,
- описание постановки на ЭВМ,
- подробное описание /для модификации программы/,
- описания и тексты СДФ-блоков.

Краткое описание адресуется пользователю - лицу, использующему программу для решения задач на ЭВМ. В нем разработчик программы описывает

- назначение программы,
- условия применимости программы,
- форму представления и структуру входных данных и результатов программы,

- практическое использование программы /например, взаимодействие с операционной системой /ОС//,

- особые ситуации и реакцию на них программы и ОС.

Краткое описание вместе с экземпляром программы на языке загрузчика /редактора связей/ либо заменяющей этот экземпляр информацией о местонахождении программы в библиотеке должны быть достаточны для решения задач с помощью описываемой программы.

Под постановкой программы на ЭВМ понимается получение краткого описания программы на ЭВМ в виде печатного материала, преобразование СДФ-программы в программу на входном языке загрузчика /редактора связей/ и проверка работоспособности последней /например, путем расчета контрольного варианта/. Описание постановки на ЭВМ должно иметь на этот счет исчерпывающую информацию.

Создавая программу, ее автор принимает решения, касающиеся выбора представления данных и алгоритмов их обработки, выделения в программе функциональных и технологических модулей /подпрограмм и СДФ-блоков/, обозначений программных объектов. Для того, чтобы создаваемую программу можно было модифицировать без непосредственных контактов с ее автором, принятые решения должны быть зафиксированы в подробном описании, содержащем:

- перечень и описание подчиненности фортран-модулей,
- перечень и описание подчиненности СДФ-блоков,
- словарь имен программных объектов,
- общую таблицу ссылок,
- описания внутренних структур данных,

и описаниях и текстах СДФ-блоков, где для каждого блока дано:

- описание блока /назначение, выполняемые действия, особые ситуации/,
- текст блока на псевдокоде,
- текст блока на СДФ.

Подробное изложение структуры перечисленных выше компонент СДФ-программы выходит за рамки настоящей работы.

7. О ПРАКТИЧЕСКОМ ПРИМЕНЕНИИ СДФ

Становление и развитие СДФ шло одновременно с его практическим применением при создании комплекса программ обработки гравиметрических данных /9/ и транслятора "СДФ - фортран". В обеих разработках помимо познаваемости достигнута мобильность программного обеспечения за счет использования подмножества фортрана, общего для его диалектов, принятых на ЭВМ центрального вычислительного комплекса ОИЯИ /БЭСМ-6, CDC-6500, ЕС-1060/.

Автор благодарен В.П.Ширикову за конструктивные замечания.

ЛИТЕРАТУРА

1. Корнейчук А.А. ОИЯИ, 11-80-382, Дубна, 1980.
2. Дал У., Дейкстра Э., Хоор К. Структурное программирование. "Мир", М., 1975.
3. Йодан Э. Структурное проектирование и конструирование программ. "Мир", М., 1979.
4. Хьюз Дж., Мичтом Дж. Структурный подход к программированию. "Мир", М., 1980.
5. Дзержинский Ф.Я., Тер-Сааков А.П. Технология программирования - структурный подход /учебно-методическое пособие/. ЦНИИАтоминформ, М., 1979.
6. Meissner L.P. On Extending FORTRAN Control Structures to Facilitate Structured Programming. SIGPLAN Notices, 1975, Sept., pp.19-30.
7. Вельбицкий И.В., Ходаковский В.Н., Шолмов Л.И. Технологический комплекс производства программ на машинах ЕС-ЭВМ и БЭСМ-6. "Статистика", М., 1980.
8. Брукс Ф.П. Как проектируются и создаются программные комплексы. "Наука", М., 1979.
9. Корнейчук А.А., Литвиненко О.К. ОИЯИ, 10-12534, Дубна, 1979.
10. Корнейчук А.А. В сб.: Современные методы и средства программирования. МДНТП, М., 1981, с.98-103.
11. Gade B. Text Formatting Program (BARB). CERN Program Library, Q500.
12. Салтыков А.И., Семашко Г.Л. Программирование для всех. "Наука", М., 1980.
13. FORTRAN Version 5 Reference Manual. Control Data Corporation, Publication No.60481300, 1980.
14. Дзержинский Ф.Я. Псевдокод /язык проектирования программ/ и правила структурного программирования на фортране /учебно-методическое пособие/. ЦНИИАтоминформ, М., 1979.
15. UPDATE Reference Manual. Control Data Corporation. Publication No.60449900, 1975.
16. Воеводин В.В., Гайсарян С.С., Кабанов М.И. В сб.: Совещание по программированию и математическим методам решения физических задач. ОИЯИ, Д10-7707, Дубна, 1974, с.414-421.
17. Безбородов Ю.М. Индивидуальная отладка программ. "Наука", М., 1982.
18. Marinescu D.C. Structured Programming. JINR, E10-9938, Dubna, 1976.
19. Gajewski J. ALFOR-Structured FORTRAN Dialect and its Pre-Processor with Macro and Instruction Generation Options. PHE 77-4, Institut für Hochenergiephysik, Berlin-Zeuthen, DDR, 1977.
20. Орлов Б.Н., Семин Н.Н. Программирование, 1980,3, с.68-74. Рукопись поступила в издательский отдел 6 мая 1982 года.

НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

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

D13-11182	Труды IX Международного симпозиума по ядерной электронике. Варна, 1977.	5 р. 00 к.
D17-11490	Труды Международного симпозиума по избранным проблемам статистической механики. Дубна, 1977.	6 р. 00 к.
D6-11574	Сборник аннотаций XV совещания по ядерной спектроскопии и теории ядра. Дубна, 1978.	2 р. 50 к.
D3-11787	Труды III Международной школы по нейтронной физике. Алушта, 1978.	3 р. 00 к.
D13-11807	Труды III Международного совещания по пропорциональным и дрейфовым камерам. Дубна, 1978.	6 р. 00 к.
	Труды VI Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1978 /2 тома/	7 р. 40 к.
D1,2-12036	Труды V Международного семинара по проблемам физики высоких энергий. Дубна, 1978	5 р. 00 к.
D1,2-12450	Труды XII Международной школы молодых ученых по физике высоких энергий. Приморско, НРБ, 1978.	3 р. 00 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.
Pf8-82-117	Труды IV совещания по использованию новых ядерно-физических методов для решения научно-технических и народнохозяйственных задач. Дубна, 1981.	3 р. 80 к.

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

Корнейчук А.А.

5-82-326

О программировании на структурном диалекте фортрана

Обсуждаются проблемы применения структурного диалекта фортрана /СДФ/ в качестве инструментального средства разработки познаваемых программ, то есть таких программ, которые могут быть использованы для проведения вычислений, поставлены на ЭВМ и модифицированы без непосредственных контактов с их авторами.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1982

Kornejchuk A.A.

5-82-326

On Programming by Structured Dialect of FORTRAN Language

Some problems of Structured Dialect of FORTRAN (SDF) language application as an instrumental means of readable programs (i.e., such programs which can be used for computation, implemented on a computer and modified without direct contacts with its authors) design are discussed.

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

Preprint of the Joint Institute for Nuclear Research. Dubna 1982

Перевод автора.