

## Переход от эталонного представления языка программирования к конкретному языку

© В.А. Фисун<sup>1,2</sup>

<sup>1</sup> Институт прикладной математики им. М.В. Келдыша РАН, Москва, 125047, Россия

<sup>2</sup> МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

*Изложены особенности использования комментариев специального назначения в алгоритмических языках для адаптации программ к вычислительной среде. Рассмотрены системы параллельного программирования, основанные на использовании спецкомментариев, языки Фортран HPF, Фортран DVM и система OpenMP.*

**Ключевые слова:** *псевдокомментарии, Алгол, Фортран HPF, Фортран DVM, OpenMP.*

После появления в 60-х годах прошлого века алгоритмических языков высокого уровня — Алгол и Фортран — началась эра всеобщей формализации вычислительных алгоритмов на этих языках. Особенно удобно это было делать на языке Алгол-60 [1] с его аппаратом рекурсивных процедур, динамических массивов и математическим стилем конструкций языка. Многочисленные периодические журналы выпускали описания программ вычислительных алгоритмов из различных областей науки и техники, собирались коллекции избранных алгоритмов в виде пакетов программ и библиотек. Этот фонд алгоритмов и программ востребован до сих пор.

Описание вычислительного алгоритма на алгоритмическом языке должно быть строгим и не допускать двусмысленного толкования. Эпиграф к описанию языка Алгол-60 гласит: «То, что вообще может быть сказано, может быть сказано ясно, а о чем невозможно говорить — о том следует молчать». Однако при подготовке программ возникает необходимость в дополнительных языковых средствах для документирования и редактирования текстов программ. Наконец, при погружении программы в конкретную вычислительную среду желательно учитывать особенности архитектуры вычислителя.

В стандарте языка Алгол-60 для таких целей кроме эталонного представления языка предусмотрены еще две формы: для публикации и конкретное. Для транслитерации различных представлений языка широко используется концепция уточнения конструкций эталонного языка спецкомментариями, которые позволяют сохранять корректность программы с точки зрения эталонного стандарта языка и в то же время расширяют его возможности.

В языках программирования допускается использование комментариев, которые произвольно могут размещаться в любом месте программы. Этот аппарат применяется для неформального описания реализуемого алгоритма, указания ссылок на авторов и т. д. Комментарии служат подсказкой программисту, и они «невидимы» (прозрачны) для транслятора. Однако существует особый вид комментариев, включаемых в тексты программ, — специальные комментарии. Тексты таких комментариев начинаются с зарезервированной комбинации символов, распознаваемой специальными трансляторами, для которых эти псевдокомментарии являются директивами для исполнения.

Специальные комментарии можно применять при подготовке текстов программ для добавления в текст макроопределений, аннотаций, дополнения текста возможностями библиотек, а также для работы с уже существующей программой, например, для получения документации или для верификации программы. Популярными процедурами на уровне языка публикаций, основанного на спецкомментариях, являются системы автоматического документирования Doxygen, Epydoc, Javadoc, PHPdoc и др. Так, система Doxygen может готовить полную документацию к программе по ее исходному тексту и набору спецкомментариев. Спецкомментариями могут служить и некоторые другие виды комментариев. В языке Java комментарий TODO помещается в нереализованный участок текста программы как указание о том, что данный фрагмент программы отсутствует. Комментарий FIXME помечает замеченную ошибку, которую можно исправить позднее, а комментарий XXX — критическую ошибку в тексте программы, без исправления которой нельзя продолжать дальнейшую работу.

Комментарии специального назначения используются также для расширения возможностей эталонных языков при учете особенностей конкретной вычислительной среды выполнения программ. Вначале это были указания о семантических особенностях программ на эталонном языке, которые невозможно получить анализом текста автоматически. Специальные комментарии в конкретных представлениях языка позволяли трансляторам генерировать более эффективный объектный код. Примером одних из первых комментариев для исполнения могут служить спецкомментарии в отечественной системе программирования ТА-2, во входном языке транслятора с языка Алгол-60 для ЭВМ класса М-20. Этими комментариями можно было помечать нерекурсивные процедуры программы, задавать максимально возможную длину динамических массивов.

Для мультипроцессорных кластеров были предложены комплексы спецкомментариев, включение которых в тексты программ с последовательного языка позволяют трансляторам получать параллельный объектный код. Наиболее развитой системой спецкомментари-

ев такого вида принято считать спецдирективы программного обеспечения фирмы CONVEX для кластера SPP1000, который является классическим примером мультипроцессорной архитектуры вычислителей с общей памятью. Язык спецдиректив кластера имеет средства для указания фрагментов программы, допускающих распараллеливание, и синхронизации выполнения параллельных процессов. Кроме обычных правил разделения данных процессов на локальные и глобальные данные в этой системе программирования можно учитывать архитектурные особенности оперативной памяти сс-NUMA кластера SPP1000. Например, обработку «близкорасположенных» сегментов данных могут выполнять процессы, расположенные на таком же уровне доступа. С этой проблемой так называемой миграции процессов по ядрам в многоядерных микропроцессорах приходится считаться в современных распределенных вычислениях. Для микропроцессоров фирмы HP кластера SPP1000 трансляторы с языков Фортран и Си учитывают спецдирективы для управления работой кэш памяти, межпроцедурной оптимизации, задания коэффициентов размножения тел циклов и т. д. Существует даже директива, позволяющая передавать системе программирования указание пользователя о вероятности перехода по метке в заданном условном операторе программы при ее выполнении. Трансляторы CONVEXа в режиме ОЗ — «Распараллелить!» по тексту последовательных программ и без использования спецдиректив могут генерировать параллельный объектный код. Конечно, техника ручного управления распараллеливанием последовательных программ позволяет получать более эффективные результаты.

Возможность автоматического распараллеливания последовательных программ ограничивается системами программирования для векторных машин и мультисистем с общей памятью, поэтому для самых массовых вычислительных кластеров — MPP-систем — были востребованы средства ручной адаптации накопленного фонда последовательных программ. Многообразие мультипроцессорных систем и отсутствие языков параллельного программирования вызывало естественный интерес к использованию для параллельного программирования конкретных расширений алгоритмических языков. Так, для программирования методом передачи сообщений для MPP-систем были предложены языки Фортран GNS и Си GNS, полученные добавлением средства описания и идентификации параллельных процессов, коммуникационных операторов в основные конструкции эталонных языков.

Самыми распространенными расширениями эталонных языков для параллельного программирования, основанными на спецкомментариях, являются языки Фортран HPF, Фортран DVM и система

OpenMP (Open Multi-Processing). Соответствующие комплексы разработаны и используются также для Си-систем программирования. Считается, что такие системы параллельного программирования позволяют использовать фонд последовательных программ, накопленных на этих языках.

Основная концепция системы программирования OpenMP состоит в организации распределенных вычислений на многопроцессорных системах с общей памятью. OpenMP — набор директив, библиотечных процедур и системных переменных, которые предназначены для программирования многопоточных приложений с помощью спецкомментариев на языках Си, Си++ и Фортран. В языке Фортран знаком спецкомментария OpenMP является префикс `!$omp`, а в Си — директива `#pragma omp`. Использование специальных ключевых символов `omp` указывает на то, что эти команды относятся к OpenMP. Таким образом, директивы `#pragma` для работы с OpenMP имеют следующий формат:

```
#pragma omp <директива>.
```

OpenMP поддерживает директивы `private`, `parallel`, `for`, `section`, `single`, `master`, `critical`, `flush`, `ordered`, `atomic` и ряд других, которые определяют механизмы распараллеливания вычислительных работ или задают конструкции синхронизации. Если программа, содержащая директивы OpenMP, скомпилирована Си, Си++, Фортран-компиляторами в режиме обработки эталонного языка с игнорированием спецкомментариев, то полученный объектный код будет выполняться как последовательный. В настоящее время практически все ведущие производители многоядерных процессоров поддерживают систему OpenMP в компиляторах на своих платформах.

Для класса распределенных вычислений разработана система параллельного программирования DVM, основанная на модели параллелизма по данным (МПД). Описываемые далее целеуказания этой разработки определены в работе [2]. В этой модели отсутствуют понятие процесса и явная передача сообщений или явная синхронизация. Согласно этой модели данные последовательной программы распределяются по узлам (процессорным элементам) вычислительного кластера, а последовательная программа преобразуется в параллельную версию программы с использованием модели передачи сообщений либо модели с общей памятью. Такая модель параллелизма освобождает программиста от рутинной и трудоемкой работы по распределению глобальных массивов на локальные массивы процессов, по управлению передачей сообщений и синхронизации доступа к общим данным. Разработчики системы DVM приводят примеры высокой эффективности реализации многих алгоритмов научно-технических расчетов с помощью МПД.

Первой попыткой использования МПД для программирования научно-технических расчетов явилась разработка языка Fortran High Performance — Фортран HPF. Директивы распределения данных последовательной программы по узлам кластера и другие параллельные директивы оформляются в виде спецкомментариев языка Фортран. Стандарт HPF разрабатывался на основе теоретических исследований и базировался на идее полной автоматизации распараллеливания вычислений и синхронизации работы с общими данными. Однако первые реализации HPF показали неэффективность языка для ряда современных методов вычислений (в частности, для нерегулярных вычислений). В следующей версии стандарта HPF была предпринята попытка ручного управления эффективностью параллельного выполнения программ. В ней определены средства распределения вычислений и спецификации общих редуцированных переменных.

Модель параллелизма DVM также базируется на МПД. Аббревиатура DVM отражает два названия модели: распределенная виртуальная память (Distributed Virtual Memory) и распределенная виртуальная машина (Distributed Virtual Mashine). Эти названия указывают на возможность адаптации модели DVM как для систем с общей памятью, так и для систем с распределенной памятью. В отличие от стандарта HPF в системе DVM не ставилась задача полной автоматизации распараллеливания вычислений и синхронизации работы с общими данными. Программист полностью управляет эффективностью выполнения параллельной программы с помощью низкоуровневых спецификаций. Высокоуровневая модель DVM не только снижает трудоемкость разработки параллельных программ, но и определяет единую формализованную базу для систем поддержки выполнения, отладки, оценки и прогноза производительности программ. Эта программа помимо описания алгоритма обычными средствами языков Си или Фортран содержит правила параллельного выполнения этого алгоритма. Правила — спецкомментарии — оформляются синтаксически таким образом, что они являются «невидимыми» для стандартных компиляторов с последовательных языков Си и Фортран и не препятствуют возможности выполнения и отладки DVM-программы как обычной последовательной программы. Компилятор переводит программу на языке C-DVM (Fortran DVM) в программу на стандартном языке Си (Фортран), расширенную функциями системы поддержки выполнения DVM-программ. Библиотека процедур системы поддержки должна входить в состав математического обеспечения вычислителя при выполнении DVM-программ, она использует для организации межпроцессорного взаимодействия стандартные коммуникационные библиотеки (MPI, PVM, Router).

Прогноз дальнейшей судьбы комментариев для исполнения, спецкомментариев как средства перехода от эталонного представления языка к языку конкретного представления следует рассматривать с учетом их места в процессе развития систем программирования.

Спецкомментарии в программах, ориентированные на последовательные ЭВМ, служат полезными дополнениями к механизмам управления процессом трансляции программ. Спецкомментарии позволяют указывать на фрагменты программ, для которых, возможно, необходимо или рекомендуется применение особых алгоритмов генерации объектного кода. Так как эти алгоритмы существенно машинно-ориентированные, то унификация и стандартизация таких спецкомментариев затруднительна, однако они всегда будут востребованы для оптимизации объектного кода.

Спецкомментарии уровня поддержки параллельного программирования позволяют получить параллельный объектный код (с возможностью выполнять свою последовательную программу на параллельных вычислителях) по тексту программы на эталонном последовательном языке (с возможностями отладки, модификации и сопровождения одного варианта текста программы). Впервые такие удобства обеспечили спецкомментарии для векторных вычислителей — от С-120 до Сray-машин. Однако эффект этого аппарата спецкомментариев скрадывали встроенные в трансляторы системы векторизации. Основанные на алгоритмах Лэмпорта и Кука векторизаторы [3] могли находить и учитывать большинство векторных операций в программе при трансляции, а доля ручной доработки программы с помощью спецкомментариев была невелика.

Программы на последовательных языках для многопроцессорных кластеров превращались (а большинство и сейчас превращаются) в параллельные путем внесения в тексты программ после их реструктуризации директив обмена сообщениями (в основном, системы MPI) между процессами. До появления специализированных систем и языков адаптация последовательных программ к кластерным архитектурам вычислителей остается основным средством параллельного программирования, а одним из таких методов адаптации является использование спецкомментариев. Наиболее успешной работой в этой области была автоматизация программирования для мультипроцессорных систем с общей памятью. Спецкомментарии системы программирования кластера SPP1000 были унифицированы, дополнены, стандартизованы и объявлены системой параллельного программирования OpenMP. Размещение в последовательных программах спецкомментариев OpenMP позволяет получать параллельный объектный код для вычислительных систем с общей памятью. Проекты

Фортран HPF и DVM — дополнения соответствующими спецкомментариями эталонного языка программирования для параллельного программирования многопроцессорных кластеров матричной архитектуры. Применение спецкомментариев данного класса переводит последовательную программу в ранг параллельной программы с сохранением единого текста и с предположением об эквивалентности результатов параллельных и последовательных вычислений.

В свое время В. В. Воеводин [4], говоря о стиле программирования, приводил пример простого векторного вычисления тремя вариантами алгоритмически эквивалентных циклических конструкций, которые различаются временем выполнения на порядки. Диапазоны вариаций времени счета OpenMP программ и ее элементов, объем используемой оперативной памяти могут расширяться. Поэтому для эффективного управления динамикой выполнения программы системой спецкомментариев определены дополнительные механизмы. Однако использование в параллельной программе возможностей полного аппарата спецкомментариев OpenMP не позволяет в ряде случаев сохранять ее эквивалентность последовательной программе. Поэтому предположение о возможности возвращения к последовательному варианту параллельной программы для ее отладки и тем более о сопровождении только одного варианта программы следует считать преувеличением возможностей системы OpenMP. Наконец, система OpenMP обеспечивает параллельное программирование только для мультипроцессорных систем с общей памятью и не применима к матричным вычислителям. Система спецкомментариев DVM имеет средства настройки для оптимального управления динамикой выполнения параллельной версии программы на уровне систем интерпретации, а не на уровне эталонного языка. Так как этапу внесения в текст последовательной программы спецкомментариев предшествует работа по реструктуризации программы для параллельного выполнения, то разрыв между алгоритмически эквивалентными версиями программы увеличивается еще больше. Параллельная версия последовательной программы, полученная с помощью спецкомментариев, — всегда другая программа. Это не удивительно, так как даже трансляция последовательной программы с включенным режимом оптимизации приводит иногда к генерации объектного кода, результаты счета которого могут не совпадать с результатами счета кода, полученным без оптимизации.

Система спецкомментариев имеет непосредственное отношение к эволюции языков программирования. Практика внедрения языков программирования с широкими новыми возможностями таких, как Алгол-68, Ада не получила развития, в частности, из-за большого разрыва между уровнями классических и обобщенных языков для

большинства пользователей. Идеологи языка Фортран [5] выбрали путь последовательного совершенствования эталонного языка, его поэтапной модификации: отказа от устаревших конструкций, добавления в него новых средств для объектно-ориентированного и параллельного программирования, разделения языка на базовую часть и динамическую оболочку. Так, одна из последних версий языка включает возможности параллельного программирования для многопроцессорных систем с общей памятью. Аппарат спецдиректив может служить средством апробации новых приемов программирования, причем выдержавшая контроль система спецдиректив может претендовать на переход в основные конструкции эталонного языка. Итерационный путь создания языка, отвечающего будущим вызовам (сейчас известно не менее шести версий эталонного языка Фортран [5]) означает, что при появлении новых версий языка и соответствующих систем программирования следует поддерживать программы, написанные на старых версиях. Для этого также может быть использована система спецкомментариев. Возможно, именно концепция спецкомментариев (по примеру OpenACC [6]) позволит демпфировать будущие вызовы, таких как обеспечение программирования графических процессоров.

## ЛИТЕРАТУРА

- [1] *Алгоритмический язык АЛГОЛ-60. Пересмотренное сообщение*. Москва, Мир, 1967, 77 с.
- [2] *Языки программирования C-DVM и Fortran-DVM*. URL: <http://kiam.ru/DVM> (дата обращения 12.05.2013).
- [3] Анисков А.В., Пономаренко О.Н., Фисун В.А. Конверторы – векторизаторы. *Языки и параллельные ЭВМ. Алгоритмы и алгоритмические языки*. Москва, Наука, 1990, с. 38–61.
- [4] Воеводин В.В. *Вычислительная математика и структура алгоритмов*. Москва, Из-во МГУ, 2006, 112 с.
- [5] Горелик А.М. *Новый стандарт языка Фортран (Фортран 2008)*. ИПМ им. М.В. Келдыша. Препринт. Москва, 2011, № 16, 29 с.
- [6] *OpenACC Directives for Accelerators*. URL: <http://OpenACC-standard.org> (дата обращения 12.05.2013).

Статья поступила в редакцию 15.05.2013

Ссылку на эту статью просим оформлять следующим образом:

Фисун В.А. Переход от эталонного представления языка программирования к конкретному языку. *Инженерный журнал: наука и инновации*, 2013, вып. 4. URL: <http://engjournal.ru/catalog/mathmodel/hidden/675.html>

Фисун Владислав Андреевич — канд. физ.-мат. наук, вед. науч. сотруд. Института прикладной математики им. М.В. Келдыша РАН, доц. кафедры «Прикладная математика» МГТУ им. Н.Э. Баумана. e-mail: [fis@a5.kiam.ru](mailto:fis@a5.kiam.ru)