

Смешанное программирование на Фортране и Си

А.М. Горелик

Аннотация. В статье рассматриваются средства языка Фортран 2003/2008, которые обеспечивают взаимодействие программ, написанных на Фортране и Си. Механизм, который обеспечивает удобный интерфейс при смешанном программировании, учитывает различия в типах данных, переменных, глобальных данных и соглашениях при вызовах процедур.

Ключевые слова: Фортран, Си, стандарты, языки программирования.

Введение

В программах, написанных на Фортране, в ряде случаев возникает необходимость использовать функции, написанные на языке Си. В других ситуациях может потребоваться использовать Фортран процедуры в Си программе.

Для того чтобы прикладная программа в таких случаях работала корректно, необходимо учитывать различия в этих языках: различия в типах данных, переменных, глобальных данных и соглашениях при вызовах процедур.

Обеспечить удобный интерфейс в смешанном программировании с помощью имевшихся ранее в Фортране средств не всегда возможно. В некоторых реализациях были введены дополнительные механизмы, облегчающие взаимодействие, но эти средства не являются стандартными.

В стандарте языка Фортран (начиная с Фортрана 2003 [1, 2]) предусмотрен механизм взаимодействия с Си [3]. В частности, в Фортране предусмотрены встроенный модуль ISO_C_BINDING, атрибут BIND(C) и введены перечисленные ниже средства, которые обеспечивают удобный интерфейс Фортран программ с программами, написанными на языке Си:

- механизм установления соответствия типов данных Си и типов Фортрана;

- средства установления соответствия переменных;

- механизм установления соответствия при вызове Си функции и передаче аргументов, а также при вызове Фортран процедур из Си программы;

- средства, позволяющие устанавливать соответствие внешних имен Си и глобальных имен Фортрана.

В данной статье приводится описание средств взаимодействия с Си, предусмотренных в стандартах языка Фортран 2003/2008. Полное описание приведено в [1, 2, 4]. Описания и обзор стандартов языка Фортран приведены в [5-9].

1. Интероперабельность

Любые вызываемые объекты в смешанном программировании должны быть такими, чтобы в обоих языках для них имелись эквивалентные объявления и чтобы они одинаково интерпретировались в этих языках. Это требуется для типов, переменных и процедур.

Объекты в Фортран программе, которые могут быть использованы и в Си программе (совместимые), в стандарте языка называются *интероперабельными* (interoperable). В язык введены средства, которые позволяют компи-

лятору на этапе компиляции определить, является ли объект интероперабельным.

2. Соответствие типов

2.1. Встроенные типы

Рассмотрим сначала средства установления соответствия встроенных типов.

В языке Си predetermined типы могут иметь дополнительные спецификации (short, long, unsigned). Например, long int. Эти спецификации определяют способ представления данных в памяти.

В Фортране для всех встроенных типов имеются средства параметризации, т.е. каждый

тип может иметь несколько разновидностей, которые задаются с помощью параметров типа.

Например, INTEGER (4) или INTEGER (KIND=4).

Для того чтобы установить соответствие типов Фортрана с типами Си, используется встроенный модуль ISO_C_BINDING, который делает доступными именованные константы, используемые в качестве параметров типа. Чтобы этот модуль был доступен программе, используется оператор USE.

В приведенной ниже таблице показано соответствие типов Фортрана с параметрами типов (в виде именованных констант целого типа) и типов Си.

Соответствие типов Фортрана и Си

Тип Фортрана	Параметр типа (именованная константа)	Си тип(ы)
INTEGER	C_INT	int
INTEGER	C_SHORT	short int
INTEGER	C_LONG	long int
INTEGER	C_LONG_LONG	long long int
INTEGER	C_SIGNED_CHAR	signed char, unsigned char
INTEGER	C_SIZE_T	size_t
INTEGER	C_INT8_T	int8_t
INTEGER	C_INT16_T	int16_t
INTEGER	C_INT32_T	int32_t
INTEGER	C_INT64_T	int64_t
INTEGER	C_INT_LEAST8_T	int_least8_t
INTEGER	C_INT_LEAST16_T	int_least16_t
INTEGER	C_INT_LEAST32_T	int_least32_t
INTEGER	C_INT_LEAST64_T	int_least64_t
INTEGER	C_INT_FAST8_T	int_fast8_t
INTEGER	C_INT_FAST16_T	int_fast16_t
INTEGER	C_INT_FAST32_T	int_fast32_t
INTEGER	C_INT_FAST64_T	int_fast64_t
INTEGER	C_INTMAX_T	intmax_t
INTEGER	C_INTPTR_T	intptr_t
REAL	C_FLOAT	float
REAL	C_DOUBLE	double
REAL	C_LONG_DOUBLE	long double
COMPLEX	C_FLOAT_COMPLEX	float_complex
COMPLEX	C_DOUBLE_COMPLEX	double_complex
COMPLEX	C_LONG_DOUBLE_COMPLEX	long double_complex
LOGICAL	C_BOOL	_Bool
CHARACTER	C_CHAR	char

Например, типу `long int` языка Си соответствует тип `INTEGER (C_LONG)`.

Примечания

1. Поскольку стандарт Си специфицирует, что представление `unsigned` целых такое же как положительных целых со знаком, можно использовать для `unsigned` целых те же константы модуля `ISO_C_BINDING`, которые используются для положительных целых. В связи с этим было решено, не вводить в Фортран типы со спецификацией `unsigned`.

2. Процессор может не поддерживать некоторые из спецификаций типов Си, в этом случае значение параметра типа (именованной константы) отрицательно.

3. Для символьного типа Фортрана интероперабельность требует, чтобы значение параметра длины было равным единице (т.е. длина должна быть опущена или специфицирована в объявлении типа со значением единица).

Встроенный модуль `ISO_C_BINDING` содержит также следующие именованные константы (все символьного типа длиной равной единице).

<code>C_NULL_CHAR</code>	нулевой символ
<code>C_ALERT</code>	звонок
<code>C_BACKSPACE</code>	возврат на одну позицию
<code>C_FORM_FEED</code>	прогон листа
<code>C_NEW_LINE</code>	новая строка
<code>C_CARRIAGE_RETURN</code>	перевод строки
<code>C_HORISONTAL_TAB</code>	горизонтальная табуляция
<code>C_VERTICAL_TAB</code>	вертикальная табуляция

Встроенный модуль содержит и некоторые другие объекты, о которых речь пойдет ниже.

2.2. Производные типы

Типу структура (`struct`) языка Си в Фортране соответствует производный тип. Для того чтобы процессор отобразил структуру, аналогичную Си, т.е. чтобы тип был интероперабельным, в Фортране в описании производного типа должен быть атрибут `BIND(C)`. Каждый компонент такого производного типа должен иметь интероперабельный тип и интероперабельный параметр типа; компонент не может быть указателем или динамически размещаемым объектом.

Пример

```
USE ISO_C_BINDING
TYPE, BIND (C) :: FTYPE
  INTEGER (C_INT) :: J1, J2
  REAL (C_FLOAT) :: R12
END TYPE FTYPE
```

Описанному в примере типу `FTYPE` соответствует следующая структура в Си:

```
typedef struct {
  int j1, j2;
  float r12;
} ctype;
```

Интероперабельный (совместимый) тип должен иметь одинаковое число компонентов с соответствующей структурой Си. Соответствие компоненты устанавливается по их позиции в описании типа. Тип каждого Фортран компонента должен быть интероперабельным с соответствующим Си компонентом.

Для интероперабельности имена типа и имена компонентов несущественны.

Для типа `union` (объединение) языка Си нет прямого аналога в Фортране, однако в ряде случаев можно добиться тех же целей с помощью оператора `EQUIVALENCE`.

2.3. Тип перечисление (enum)

Перечисление (enumeration) – это набор констант целого типа (enumerators), которые соответствуют типу `enum` языка Си. Для объявления перечисления используется конструкция, которая начинается с оператора `ENUM` и заканчивается оператором `END ENUM`. Между ними помещается один или несколько операторов `ENUMERATOR`.

Пример

```
ENUM, BIND (C)
  ENUMERATOR :: WHITE = 3, BLACK = 7
  ENUMERATOR PINK
END ENUM
```

В этом фрагменте объявляются именованные константы `WHITE`, `BLACK` и `PINK` со значениями 3, 7 и 8 соответственно.

Если значение не специфицировано, оно берется на единицу больше значения предыдущей константы или равно нулю, если оно первое.

3. Указатели

Что касается типа указатель, то следует отметить, что в отличие от Си в Фортране указатель не является типом, это атрибут.

Для того чтобы установить соответствие с объектами Си, имеющими тип указатель, в Фортране предусмотрены производные типы (интероперабельные с Си): `C_PTR` – указатель на переменную и `C_FUNPTR` – указатель на функцию Си. Объекты типа `C_PTR` и `C_FUNPTR` имеют компоненты `PRIVATE` и должны быть доступными в модуле `ISO_C_BINDING`.

Этот модуль должен содержать также константы и процедуры, перечисленные ниже.

Именованные константы, соответствующие пустым указателям Си:

```
C_NULL_PTR
C_NULL_FUNPTR
```

В некоторых случаях для работы с Си функциями требуется адрес объекта. Для этого в Фортране введены следующие встроенные процедуры:

`C_LOC (X)` – справочная функция, возвращающая скаляр типа `C_PTR`: адрес аргумента `X` в Си.

`C_FUNLOC (X)` – справочная функция, возвращающая адрес функции в Си.

`C_ASSOCIATED (C_PTR1 [, C_PTR2])` – справочная функция, возвращающая статус связанности указателя.

`C_F_POINTER (CPTR, FPTR [, SHAPE])` – связывает указатель данных Фортрана с адресатом Си указателя.

`C_F_PROCPTR (CPTR, FPTR)` – связывает процедурный указатель Фортрана с адресатом Си указателя функции.

`C_SIZEOF(X)` – размер `X` в байтах.

Замечания.

Размещенный динамический массив передавать в Си можно.

Массив, размещенный в Си, может быть связан с указателем Фортрана.

В Си нельзя передавать адресат указателя Фортрана или массив, перенимающий конфигурацию фактического аргумента (`assumed shape array`).

4. Соответствие переменных

Скалярная переменная Фортрана интероперабельна с Си, если интероперабельны ее тип и параметры типа и она не является указателем или размещаемой переменной.

Переменная-массив интероперабельна, если интероперабельны ее тип и параметры типа и при этом массив является массивом с явной конфигурацией (`explicit shape`) или массивом, перенимающим размер фактического аргумента (`assumed size`). Массив соответствует массиву языка Си с тем же типом и теми же параметрами типа и конфигурацией, но с обратным порядком индексов.

Пример

Массив, объявленный в Фортране следующим образом

```
REAL, DIMENSION :: ARF(2:10, 20, *)
```

интероперабелен с массивом Си, объявленным так:

```
int arf [ ] [20] [9]
```

В смешанном программировании следует учитывать, что в Фортране массивы хранятся в памяти “по столбцам”, а в Си – “по строкам”.

5. Соответствие при вызове процедур

Для процедур, которые могут связываться с Си функциями, используется атрибут `BIND(C)`. Этот атрибут специфицирует, что из функции (`function`) или подпрограммы (`subroutine`), написанной на Фортране и имеющей данный атрибут, может быть вызвана Си функция.

Примеры.

```
FUNCTION FN (ARG1, ARG2), BIND (C)
```

```
SUBROUTINE SUB1 (ARG3), BIND (C)
```

Процедура интероперабельна, если она имеет явный интерфейс, атрибут `BIND(C)` в объявлении, все формальные аргументы интероперабельны (и не имеют атрибута `OPTIONAL`), а для функции, кроме того, интероперабельный скалярный результат функции. Интероперабельная процедура может быть внешней, модульной, но не внутренней.

Подпрограмме Фортрана соответствует функция Си с типом `void`, а функции Фортрана

соответствует функция Си, имеющая тип, отличный от void.

Фортран (до Фортрана 2003), как известно, определял способ передачи аргументов по ссылке, в Си параметры могут передаваться по значению. Для того чтобы обеспечить возможность взаимодействия для формальных аргументов, в операторах объявления типа введен атрибут VALUE (и соответственно оператор VALUE).

Формальный аргумент с таким атрибутом соответствует формальному параметру Си не ссылочного типа, а формальный аргумент Фортрана без атрибута VALUE соответствует формальному параметру Си, который является указателем.

Очевидно, что атрибут VALUE можно использовать только для скаляров, не имеющих спецификации INTENT(OUT) или INTENT(INOUT). Аргументы с атрибутом VALUE не должны быть указателями и размещаемыми. Для символьного типа длина таких аргументов должна быть равна единице.

Для того чтобы имя Фортран процедуры (кроме модульной процедуры с атрибутом PRIVATE) было доступно Си процессору, используется *метка связывания*. Она представляет собой имя, под которым эта процедура известна процессору Си. По умолчанию это имя процедуры в Фортране в нижнем регистре. В приведенных выше примерах метка связывания может быть fn и sub1 соответственно. Метка связывания может быть указана в заголовке процедуры. Например,

```
FUNCTION FN (ARG1, ARG2), BIND (C,  
NAME = 'c_fn')
```

В этом случае функция соответствует прототипу функции Си с такой же меткой связывания. Результат функции должен быть интероперабельным с результатом прототипа. Для подпрограммы результат прототипа должен иметь тип void.

Формальный аргумент с атрибутом VALUE и типом отличным от C_PTR должен соответствовать формальному параметру прототипа, который не является указателем.

Формальный аргумент без атрибута VALUE или с этим атрибутом, но имеющим тип C_PTR

должен соответствовать формальному аргументу прототипа, который является указателем.

6. Соответствие глобальных объектов

Одной из серьезных проблем является установление соответствия имен глобальных объектов Фортрана с именами внешних объектов Си. Трудность заключается в том, что правила образования имен для двух языков имеют некоторые различия и, кроме того, возможно перекрытие имен.

Для решения этой проблемы в Фортран введены средства для отображения имен, которые позволяют Фортран процессору специфицировать имена Си функций и глобальных данных.

Решение этой проблемы для процедур описано выше в предыдущем разделе. Аналогично решается вопрос для COMMON-объектов и PUBLIC-объектов, объявленных в программной единице-модуле. Для указанной цели используется атрибут BIND(C) и метка связывания (определяемая по тем же правилам, которые действуют для процедур), которая соответствует Си-переменной соответствующего типа.

Заключение

Дальнейшее развитие средств взаимодействия Фортрана и Си

Описанные выше средства взаимодействия программ на Фортране и Си имеют пока некоторые ограничения, отмеченные по ходу изложения. В частности, для взаимодействия процедурных интерфейсов в стандарте допускаются следующие аргументы: скаляр, не имеющий атрибута OPTIONAL, массив явной конфигурации (explicit shape array), массив, перенимающий размер фактического аргумента (assumed size array).

Не обеспечивается совместимость для таких аргументов: массив, перенимающий конфигурацию фактического аргумента (assumed shape array), объект символьного типа, перенимающий длину фактического аргумента (assumed character length), аргументы, имеющие атрибут ALLOCATABLE, POINTER или OPTIONAL.

В настоящее время подготовлен документ “Technical Specification” [10], который расширяет возможности Фортрана 2003/2008 по взаимодействию Фортрана и Си и снимает имеющиеся ограничения. В Фортран вводятся новые понятия *assumed rank* и *assumed type* и вводятся новые встроенные процедуры.

Следует отметить, что не требуется, чтобы были сделаны какие-либо изменения в стандарте языка Си.

Указанные расширения предполагается включить в следующий стандарт, который имеет пока рабочее название Fortran 2015.

Литература

1. ISO/IEC 1539-1: 2004. Information technology – Programming languages — Fortran – Part 1: Base Language.
2. ISO/IEC 1539-1: 2010. Information technology – Programming languages — Fortran – Part 1: Base Language.
3. ISO/IEC 9899:1999, Programming languages – C.
4. Metcalf M., Reid J., Cohen M. Fortran 95/2003 Explained. Oxford.
5. Горелик А.М. Программирование на современном Фортране. М.: Финансы и статистика, 2006 С.350.
6. Горелик А.М. Эволюция языка программирования Фортран (1957–2007) и перспективы его развития. // Вычислительные методы и программирование, 2008, т.9, №2, С.223-241.
7. Горелик А.М. Сравнение стандартов языка Фортран. // Информационные технологии и вычислительные системы, 2015, №3, С.45-64.
8. Gorelik A.M. Object-Oriented Programming in Modern Fortran. // Programming and Computer Software, 2004, №3, С. 173-179.
9. Горелик А.М. Новый стандарт языка Фортран (Фортран 2008). // Препринты ИПМ им. М.В. Келдыша РАН. 2011. №16. 29 с.
10. TS 29113: “Further Interoperability of Fortran with C” ISO/IEC JTC1/SC22/WG5 N1942.

Горелик Алла Моисеевна. Старший научный сотрудник ИПМ им. М.В. Келдыша РАН. Окончила физико-математический факультет Московского областного педагогического института. Кандидат физико-математических наук. Член ISO/IEC JTC1/SC22/WG5. Автор более 100 печатных работ. Область научных интересов: языки программирования и компиляторы, технологии программирования больших вычислительных задач, параллельные вычисления. E-mail: gorelik@keldysh.ru