7

第7章

C/C++ と Fortran の混在

インテル Visual Fortran では、他の言語で記述されたサブプログラムを呼び出すことができます。すでに開発済みのコードや、Fortran では処理しにくい処理を他の言語で記述することで開発効率を改善できます。この章ではインテル Visual Fortran と Windows 上の C/C++ 言語の混在利用について説明しています。C/C++ と Fortran が混在したプログラミングでは、次のことが可能になります。

- 別の言語で書かれている既存のコードを呼び出す。
- 特定の言語では実装が難しいプロシージャを使用する。
- 処理速度を向上させる。

アセンブリ言語などの他の言語についても混在が可能ですが、ここでは主にインテル Visual Fortran と C/C++ に注目します。この 2 つの言語は、関数、サブルーチン、およびプロシージャをほぼ同じ方法で実装しているため、混在したプログラミングが比較的容易に行えます。インテル Visual Fortran をインストールすると、コンパイルおよび実行可能な言語が混在したサンプルプログラムがインストールディレクトリに作成されます。

メインプログラムからのサブプログラムの呼び出し

呼び出し規則が一致していれば、インテル Visual Fortran のメインプログラムは C/C++ で記述されたサブプログラムを呼び出すことができます。逆にインテル Visual Fortran のサブプログラムを C/C++ のメインプログラムが呼び出すことも可能です。

Microsoft Visual Studio のプロジェクトは単一の言語のみサポートするため、Fortran と C/C++ それぞれ個別のプロジェクトを作成する必要があります。

インテル Visual Fortran では、C との互換性保持のために、いくつかの Fortran 2003 の機能が提供されます。両方の言語で対応する宣言を記述できる場合は、互換性があるとみなされます。変数、派生型、プロシージャの互換性が代表的なものですが、次の機能がサポートされています。

- BIND 属性および文。オブジェクトが C と互換性があること、およびオブジェクトに外部参照があることを指定します。
- FUNCTION 文および SUBROUTINE 文の言語バインド
- 派生型文の言語バインド

Fortran と C/C++ プログラミングの注意事項

Fortran と C/C++ が混在したプログラミングのキーポイントは次のとおりです。

- Fortran と C では、関数およびルーチンの実装方法が異なりますが、C のメインプログラムから Fortran のサブルーチンを外部の void 関数として呼び出すことができます。

各言語のルーチン呼び出しの対応関係

言語	戻り値付きの呼び出し	戻り値なしの呼び出し
Fortran	FUNCTION	SUBROUTINE
C および C++	function	(void) function

- 一般に、Fortran と C が混在したプログラムは、一方の言語で記述された既存のコードを使用することを目的に作成されます。Fortran と C は互いのルーチンを呼び出すことができるので、メインルーチンはどちらの言語でも記述することができます。
- 複数の言語が混在したプログラムの作成に Visual Studio 開発環境を使用する場合は、各言語で使用する Visual Studio 開発環境が同じバージョンでなければなりません。
- Fortran は外部名に下線を追加します。C では追加しません。
- Fortran は外部名を大文字に変換します。C はオリジナルの大文字、小文字をそのまま使用します。
- Fortran は数値データを参照で渡します。C は値で渡します。

Fortran のデフォルト動作のいくつかは、ATTRIBUTES 宣言および ALIAS 宣言を使用して変更できます。ATTRIBUTES C は、Fortran の外部名および数値データの引渡しで C と同じ動作をするように設定します。ALIAS は、Fortran が外部名にオリジナルの大文字、小文字を使用するように設定します。

- Fortran サブルーチンは、C の void ルーチンに対応します。
- Fortran では文字レングスの引渡しが必要です。C は、文字列に続く NULL により文字レングスを算出できます。つまり、Fortran プログラムが C ルーチンに文字列を渡す場合、その文字列は NULL で終了しなければなりません。
  "mystring"c または StringVar // CHAR(0)
- COMPLEX、REAL*16、および CHARACTER 型では、Fortran は関数の戻り値を格納するため隠れ引数を第 1 引数として追加します。

7-1 Fortran/C 呼び出し規則

C/C++ のモジュールでは、__stdcall キーワードを関数プロトタイプとして使用することにより STDCALL 呼び出し規則を指示することができます。__stdcall は Window プロシージャや API 関数によっても使用されます。以下の C 言語のプロトタイプは STDCALL 呼び出し規則による関数呼び出しを設定します。

```
extern void __stdcall FORTRAN_ROUTINE (int n);
```

もうひとつの方法は、C コードの呼び出し規則を変更するかわりに、ATTRIBUTES 宣言子の C のオプションを使うように Fortran ソースコードを変更することです。以下の宣言は C の呼び出し規則でサブルーチンが呼び出されることを指定します。

```
SUBROUTINE CALLED_FROM_C (A)
   !DEC$ ATTRIBUTES C :: CALLED_FROM_C
   INTEGER A
```

7-1-1　命名規則

デフォルトでは、Fortran コンパイラは関数名とサブプログラム名を大文字に変換します。しかし、C コンパイラは大文字と小文字を変換せずに区別します。このため、Fortran プログラムから呼び出される C プロシージャは、大文字と小文字を正確に使用して名前を付ける必要があります。例えば、次のような呼び出しの場合を考えてみます。

CALL PROCNAME()	C プロシージャに、PROCNAME という名前を付けなければなりません。
X=FNNAME()	C プロシージャに、FNNAME という名前を付けなければなりません。

最初の呼び出しでは、PROCNAME によって返される値は無視されますが、FNNAME は値を返さなければなりません。

7-1-2　引数の受け渡し

デフォルトでは、Fortran サブプログラムは引数を参照によって渡します。つまり、Fortran