

COBOL 利用技術のご紹介

COBOL コンソーシアム利用技術分科会
高木 渉 (日立製作所 ソフトウェア事業部)

第3回 COBOL と他言語との連携 (その 1—C 言語)

1. COBOL と他言語の連携の意味

COBOL でいろいろなアプリケーションを記述することができる一方、他の言語で書かれたプログラムとの連携が必要になることがあります。たとえば、C 言語で書かれたミドルウェアを COBOL アプリケーションから呼び出して利用したり、COBOL アプリケーションを他の言語で書かれたアプリケーションが呼び出したりする場面があります。これらを実現するのが、他言語との連携機能です。

COBOL と連携する他言語は複数ありますが、まずは基本となる連携として、C 言語との連携を取り上げます。その他の言語との連携については別の機会に紹介します。

2. COBOL と C 言語の連携

C 言語との連携は、COBOL の他言語連携の中でも中心的な役割を担います。

C 言語インターフェースは、各種の UNIX 系や Windows オペレーティングシステムで基本となる呼出しインターフェースで、C 言語の関数呼出しでの制御の移し方、引数の渡し方、戻り値の戻し方などの振る舞いが、マシン命令に近いレベルで決まっています。多くの COBOL 実装は、CALL 文の振る舞いをこの C 言語インターフェースに合わせてあり、COBOL から C 言語を呼び出すことも、またその逆のこともできるようになっています。

2.1 COBOL から C を呼ぶ場合

COBOL から C 言語の関数を呼び出す場合には、次の 3 点を注意します。

- (1) 名前の対応(呼び出す先の関数名)
- (2) 引数の渡し方の対応(COBOL の CALL 文の USING 指定での、BY REFERENCE 等の指定と C 言語の引数)
- (3) 型の対応(引数の型、および、構造を持つデータのメモリレイアウト)

以下、これらについて説明します。

2.1.1 名前の対応

C 言語と COBOL では、プログラムに記述する名前(変数名、関数名など)に許される文字の構成が異なります。

- ・ COBOL のプログラム名は、PROGRAM-ID 段落に指定された名前です。COBOL からは、CALL 文にこのプログラム名を定数または一意名の値として指定して呼び出します。
- ・ C 言語では定義した関数の名前を用いて関数を呼び出します。

COBOL から C 言語の関数を呼び出すには、CALL 文を使い、C 言語の関数名を定数または一意名の値として指定して呼び出すことになります。ただし、名前を一致させるには次のような問題があり、多くの場合 COBOL 処理系がなんらかの解決方法を用意しています。使用する COBOL 処理系と C 言語処理系のマニュアルを参照してください。

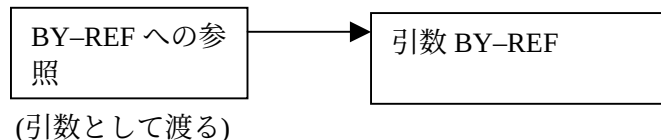
- (1) 名前に指定できる文字の種類が異なります。例えば、COBOL ではハイフンが使えますが、C 言語では使えません。
- (2) 名前の長さの最大が異なります。COBOL では 30 文字までですが、C 言語では全体文字数の制限は無く、区別すべき先頭文字数が決められています。
- (3) COBOL では大文字と小文字は等価ですが、C 言語では区別します。

2.1.2 引数の渡し方の対応

まずは、他の言語との連携ではなく、COBOL から COBOL を呼び出す際の引数の渡し方について説明します。COBOL の CALL 文の引数には、BY REFERENCE または BY CONTENT を指定して渡し方を変えることができます。

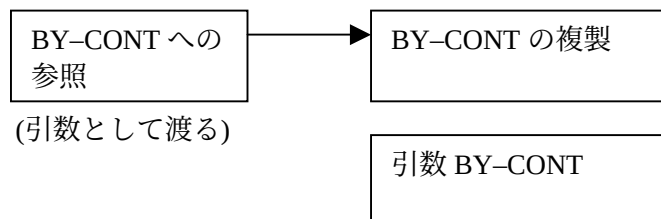
BY REFERENCE を指定した引数の場合、呼ばれるプログラムに渡すのは、値の格納された場所への参照であり「参照渡し」になります。この渡し方の特徴は、呼ぶ側と呼ばれる側で同じ格納場所を参照することです。呼ばれる側で値を変更すると呼ぶ側でもその変更が有効になります。

```
01 BY-REF PIC S9(9) BINARY.
CALL "kansuuA" USING BY REFERENCE BY-REF.
```



一方、BY CONTENT を指定した引数は、「内容渡し」になります。呼ばれる側で値を変更しても、その変更は呼ぶ側では有効になりません。内容を渡す方法としては、引数として指定されたデータ項目の値を別の場所に複製して、その場所への「参照」を、呼ばれるプログラムに渡します。こうして、呼ばれるプログラムで、BY REFERENCE と BY CONTENT を区別しなくても、共通に引数を受け取れます。受け取り方は共通ですが、BY CONTENT で渡された引数の値をいくら変更しても、呼び出し元には影響しません。

```
01 BY-CONT PIC S9(9) BINARY.
CALL "kansuuB" USING BY CONTENT BY-CONT.
```



では、COBOL の外との連携として、COBOL から C を呼ぶ場合を考えます。CALL 文の引数が BY REFERENCE でも BY CONTENT でも、結局、渡されるのは参照ですので、C 言語では、ポインタで受け取ります。このとき引数の名前は一致させる必要はありません。ポインタの先の値が本当に渡したい実体です。以下のコーディング例では、上の二つの例で制御を渡される C 言語の関数です。どちらも同じ構文になります。(型の対応については次節を参照。ここでは、COBOL の USAGE BINARY とを対応させていますが、処理系に依存します。)

BY REFERENCE で引数を渡される C 関数の例

```
void kansuuA(int *ByRef)
{
    /* 関数の処理 */
}
```

BY CONTENT で引数を渡される C 関数の例

```
void kansuuB(int *ByCONT)
{
    /* 関数の処理 */
}
```

現行規格の仕様としてはここまでですが、BY VALUE 指定についても説明します。BY VALUE 指定は、次期 COBOL 規格で導入されます。COBOL 処理系によってはすでに拡張仕様になっています。この指定の場合、引数の渡し方は「値渡し」となり、上の二つの例のように参照が渡るのではなく、値が渡ります。実際には、スタック領域などに指定された引数の値を積み、呼ばれる側ではそのままその値を受け取ります。C 言語との連携では、ポインタでなく、そのまま受け取ります。

```
01 BY-VAL PIC S9(9) BINARY.
CALL "kansuuC" USING BY VALUE BY-VAL.
```

BY-VAL の複製

(そのまま引数として渡る)

引数 BY-VAL

BY VALUE で引数を渡される C 関数の例

```
void kansuuC(int byVal)
{
    /* 関数の処理 */
}
```

引数の戻し方に関連して、戻り値についても、ここで説明します。

現在の規格 COBOL(1985 年規格)の範囲で、COBOL で呼出し元に値を返すには、BY REFERENCE を指定した引数を渡し、呼ばれる側でその内容を変更することで実現します。次期規格 COBOL では、CALL 文に明示的に RETURNING 指定を書くことで、戻り値用のデータ項目を指定できます。RETURNING 指定も COBOL 処理系によってはすでに拡張仕様になっています。

```
01 RETURN-VAL PIC S9(9) BINARY.
CALL "kansuuD" RETURNING RETURN-VAL.
```

戻り値を返す C 関数の例

```
int kansuuD()
{
    int ret;
    /* 関数の処理 */
    return ret;
}
```

さらに、COBOL での戻り値の受け取り方として、RETURN-CODE 特殊レジスタという宣言無しに使用できる特別な変数を用意している COBOL 処理系もあります。このレジスタは、たとえば、C 言語では、4 バイトの整数型に対応します。詳しくは、使用する処理系のマニュアルを参照してください。

2.1.3 型の対応

すでに上の説明で暗黙のうちに使っていますが、値を正しく引き渡すために、引数の型の対応は重要です。たとえば、

```
01 A PIC 9(9) USAGE BINARY.
```

と定義されていれば、ある実装では C 言語の 4 byte の符号なし整数に対応します。(この対応は処理系に依存します。POSIX で定義された型である、USAGE COMPUTATIONAL-5 を対応させる実装もあります。)

また、COBOL 集団項目中の基本データ項目の並べ方(レイアウト)は、C 言語の構造体メンバの並べ方と同じとは限りません。隙間バイトの位置や個数が異なります。使用する COBOL と C のそれぞれのマニュアルを見て、慎重に両者を合わせる必要があります。

個々の型の対応とレイアウトは、処理系によって異なるため、ご使用になる COBOL 処理系のマニュアルと C 言語のマニュアルを参照してください。

2.2 C から COBOL を呼ぶ場合

C から COBOL を呼ぶ場合も、COBOL から C を呼ぶ場合と同じように、次の 3 点を注意する必要があります。

- (1) 名前の対応(呼び出す先の関数名)
- (2) 引数の渡し方の対応(COBOL の手続き部見出しの USING 指定での、BY REFERENCE 等の指定と C 言語の引数)
- (3) 型の対応(引数の型、および、構造を持つデータのメモリレイアウト)

これらのうち、(1)名前の対応 と (3) 型の対応については繰り返しになりますので割愛し、(2)の引数の渡し方の対応 について説明します。

2.2.1 引数の渡し方の対応

今度は、COBOL で引数を受け取る場合の話から始めます。COBOL プログラムの手続き部見出し (PROCEDURE DIVISION header) に BY REFERENCE が指定された引数は、文字通り参照渡しで渡されることを期待しています。たとえば、

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. BYREFPROG.  
:  
LINKAGE SECTION.  
01 BY-REF PIC S9(9) USAGE BINARY.  
PROCEDURE DIVISION USING BY REFERENCE BY-REF.  
:
```

となっているとき、呼ぶプログラムの側では、渡したい値の格納されている場所(アドレス)を指す「参照」を渡す必要があります。

C から COBOL を呼ぶときは、これに合わせて、次のように変数の参照を取得してから呼び出します。

```
int ByRef;  
BYREFPROG(&ByRef); /* 関数呼出し */
```

また、次のように COBOL プログラム側が BY VALUE 指定(値渡し)で引数を待っていると
きは、

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. BYVALPROG.  
:  
LINKAGE SECTION.  
01 BY-VAL PIC S9(9) USAGE BINARY.  
PROCEDURE DIVISION USING BY VALUE BY-VAL.  
:
```

C 関数から呼ぶには、値そのものを渡す必要があります。C 言語では次のようにそのまま引数を指定すれば対応が取れます。

```
int ByVal;  
BYREFPROG(ByVal); /* 関数呼出し */
```

戻り値についても、COBOL から C を呼び出す場合と同様です。なお、先に 2.1.2 で述べたように戻り値の指定は拡張仕様ですので、詳細は処理系によります。

C から COBOL を呼ぶとき、COBOL 側の手続き部見出しに RETURNING 指定があれば、C 側では、戻り値が帰ってくるものとしてプログラミングできます。COBOL 側で RETURN-CODE 特殊レジスタに値を設定すると、C に制御が戻ったときに、その値がたとえば、整数型の関数値として利用できます。

3. まとめ

以上のように、言語仕様の差にさえ気をつければ、COBOL と C 言語との相互連携は、特別な仕組みやプログラミング無しにできます。

— 第3回 終 —