

実際にプログラムを 書いてみよう

佐藤 亨

ここでは、Visual Studio無償版を使ってサンプル・プログラムを動かしてみる。リストを実際に入力して、C言語プログラムを学んでほしい。画面表示ができない環境ならば、ブレーク・ポイントで値を確認する。サンプル・プログラムとして"Hello, World!"や数字当てゲーム、スタック的な動きをする電卓プログラムを用意した。
(編集部)

1. C言語を味わう方法

さて、C言語のお勉強にもそろそろ疲れてきたところではないですか？ ひとつここは気分転換といきましょう。前章までに学んだ知識のおさらいも兼ねて、手(とプログラム)を動かしてみましょ。

今回紹介するソース・コードはWebサイト(<http://www.cqpub.co.jp/interface/>)からすべてダウンロードできますが、もし時間(とやる気)があるならば、C言語をじっくり味わう意味でも、手で打ち込んでいただけるといいかと思います。

● 環境を作る——タダより安いものはない？！

今回は、C言語で作ったプログラムをパソコンの画面上で実行させます。C言語の開発環境は、大抵のOSに用意されているので、使っているのがWindowsやLinux、Mac OS Xでも大丈夫です。本稿では、Windows(XP/Vista)に無料の開発環境をセットアップする方法を紹介します。もし、既にC言語のプログラムを開発・実行できる環境をお持ちの場合は、以下のインストールはすっ飛ばしていただいて構いません。

今回インストールするのは、Microsoft社が提供している「Visual Studio」シリーズの一つ、「Visual C++ 2008 Express Edition」(以下VC++)です。Visual Studioにはほかにも「Standard Edition」, 「Professional Edition」など幾つかのラインナップがあって、それぞれ提供されている機能が違います。今回使用する「Express Edition」は、ほか

の製品に比べると機能がある程度限定されているのですが、無償で使えます(Microsoftさん太っ腹!)。機能限定とはいえ、C言語を体験してみるには十分すぎるほどです。もし、将来あなたがレベルアップして、Express Editionでは物足りなくなったときは、有償版を購入してください。

ところで、VC++は「C++言語」用の開発環境です。でも大丈夫。C++言語はC言語の(ほぼ)上位互換なのです。

コラム1 VC++のインストール

では、さっそくVC++をインストールしてみましょう。まず、以下のURLに行きます。

<http://www.microsoft.com/japan/msdn/vstudio/express/>

VC++は黄色いバナーのところですよ。最初はその右下にある「初めての方のためのインストール方法」を見てください。スクリーン・ショット付きで、とっても丁寧にインストール方法が書かれています。その通りにやればインストール完了です。以上。

…という説明ではちょっと寂しいので、少し補足します。まず、VC++本体でちょうど100Mバイト、オプションを含むとさらに数百Mバイトのダウンロード量になるので、それに耐えうるネット環境でセットアップを行ってください。ダウンロード後のインストールも結構な時間がかかります。もし、開発環境とネット環境が別々の場合は、直接インストールするのではなく、いったんDVD-Rに書き込んでからインストールする方法もあります。上記Webサイトの右側の方に「オフラインインストール方法」というのがあるので、そちらをご覧ください。

大抵のC++コンパイラはC言語にも対応しています。VC++は、ソース・ファイルの拡張子などを見てそれに合ったコンパイルをしてくれます。

● C言語の神様にごあいさつ

開発環境が整ったところで、いよいよプログラムを書きます。まずはC言語の神様にごあいさつします。Cプログラマなら誰でも一度は書いたことがある、例のアレです。

VC++をお使いの皆さんは、以下の手順でプロジェクトを作成してください。helloのプロジェクトなので「helloプロジェクト」です。

- (1) メニュー・バーの「ファイル」→「新規作成」→「プロジェクト」をクリック
- (2) 左側の「プロジェクトの種類」で「Win32」をクリック
- (3) 右側のテンプレートから「Win32 コンソール アプリ

コラム2 最近のコンパイラは優秀

川村 貴仁

C言語はその記述の柔軟性ゆえに、ちょっとした記述ミスがあってもエラーも出力されずにコンパイルが完了し、オブジェクトが生成されてしまうことがあります。このような記述ミスは発見が困難で、不具合の原因となってしまいます。

こういった問題を防ぐには、開発者の綿密なチェックや、コーディング・ルールによるあいまいな記述の遮断などが効果的です。しかし、開発者も人間ですので、体調の悪化や納期切迫による焦りなどの理由でチェック・ミスをすることがあります。こういったミスを防ぐため、最近のコンパイラでは、よくある記述ミスは検出できるようになっています。

コンパイラで検出可能なミスの例を以下に示します。

1) 条件判断における等値演算子(==, !=)の記述ミス

if文で条件判断を“==”ではなく“=”で記述してしまった場合、「warning: use of "=" where "==" may have been intended」のように確認メッセージが出力されます。

2) ゼロ除算

定数などで除算するコードを記述したときに、除数がゼロになってしまった場合、「warning: division by zero」のようにゼロで割っていることを警告するメッセージが出力されます。

3) 変数の範囲チェック

変数への数値代入時に、変数の型の範囲から外れている場合、「warning: integer conversion resulted in truncation」のように型変換の範囲を超えたことを示す警告が出力されます。

リスト A 無意味な switch 文の例

```
u1_Test1 = 3;
switch (u1_Test1) {
    case 1:
        u2_Test2 = U2_Ex_MAX;
        break;
    case 2:
        u2_Test2 = U2_Ex_MAX;
        break;
}
```

リスト B ゼロ除算の例

```
u1_Test2 = 3;
u1_Test1 = u1_Test1 / (u1_Test2 - u1_Test2);
```

リスト C 配列を使った例

```
for (i = 0; i < 5; i++) {
    u1_array[i] = i;
}
```

このほかにもコンパイラによっては、シミュレータに連動した形でランタイム・エラーを検出する機能を持っているものもあります。このようなコンパイラでは、以下のような問題を検出可能です。

a) switch 文の条件を超えた場合のチェック

switch文でリストAのようなソースを記述した場合、u1_Test1には3が代入されており、case文で指定された条件のいずれにも一致しないため、switch文の意味がない可能性があります。このような事例の場合、ランタイム・チェックの結果として「Stopped by runtime error: Case/switch index out of bounds on line xx of procedure」という警告が出力されます。

b) 変数によるゼロ除算のチェック

リストBのようなソースを記述した場合、コンパイル時には演算結果がどうなるかは決まっていなかったので判断できないため、ランタイム・エラー・チェックの結果、「Stopped by runtime error: Divide by 0 on line xx of procedure Check_Test1」といったエラーが出力されます。

c) 配列の境界を越えた代入のチェック

リストCのような例の場合、配列への格納先を示す数値に変数iが使用されているため、コンパイル時にはiのとりうる値と配列のサイズの関係を判断できません。そこで、実行時のランタイム・チェックで判断することになります。その結果として「Stopped by runtime error: Array index out of bounds on line xx of procedure」という警告が出力されます。

このように、最新のコンパイラでは記述ミスや演算ミスによる不具合の混入を防ぐような仕組みがあります。積極的に利用してみたいかがでしょうか。